



Universidade Estadual de Campinas  
Instituto de Computação



Matheus Silva Mota

LinkedScales: Multiscaling a Dataspace

LinkedScales: Bases de Dados em Multiescala

CAMPINAS  
2017

**Matheus Silva Mota**

**LinkedScales: Multiscaling a Dataspace**

**LinkedScales: Bases de Dados em Multiescala**

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

**Supervisor/Orientador: Prof. Dr. André Santanchè**

Este exemplar corresponde à versão final da Tese defendida por Matheus Silva Mota e orientada pelo Prof. Dr. André Santanchè.

CAMPINAS  
2017

**Agência(s) de fomento e nº(s) de processo(s):** CAPES; CNPq, 141353/2015-5

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Ana Regina Machado - CRB 8/5467

M856L Mota, Matheus Silva, 1986-  
LinkedScales : multiscaling a dataspace / Matheus Silva Mota. – Campinas, SP : [s.n.], 2017.

Orientador: André Santanchè.  
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Integração de dados (Computação). 2. Bancos de dados. 3. Banco de dados - Gerência. 4. Multiescala. I. Santanchè, André, 1968-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** LinkedScales : bases de dados em multiescala

**Palavras-chave em inglês:**

Data integration (Computer science)

Databases

Database management

Multiscale

**Área de concentração:** Ciência da Computação

**Titulação:** Doutor em Ciência da Computação

**Banca examinadora:**

André Santanchè [Orientador]

Debora Pignatari Drucker

Juan Manuel Adán Coello

Ivan Luiz Marques Ricarte

Ariadne Maria Brito Rizzoni Carvalho

**Data de defesa:** 18-12-2017

**Programa de Pós-Graduação:** Ciência da Computação



Universidade Estadual de Campinas  
Instituto de Computação



**Matheus Silva Mota**

**LinkedScales: Multiscaling a Dataspace**

**LinkedScales: Bases de Dados em Multiescala**

**Banca Examinadora:**

- Prof. Dr. André Santanchè  
Instituto de Computação – Universidade Estadual de Campinas
- Dra. Debora Pignatari Drucker  
Embrapa Informática Agropecuária
- Prof. Dr. Juan Manuel Adán Coello  
Pontifícia Universidade Católica de Campinas
- Prof. Dr. Ivan Luiz Marques Ricarte  
Faculdade de Tecnologia – Universidade Estadual de Campinas
- Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho  
Instituto de Computação – Universidade Estadual de Campinas

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 18 de dezembro de 2017



# Dedictory

*I dedicate this work to my inspiring uncle Jorge,  
the first in our family to explore the scientific world;  
example of determination, simplicity, and courage.*

# Acknowledgments

First and foremost, I would like to thank Professor André Santanchè, for changing my life throughout so many opportunities, for the countless lessons that made me grow professionally and personally, for the friendship, for believing in my potential, and for all time and patience dispensed to me in order to teach me the art of scientific research.

I would like to thank my beloved parents, Olga and Antônio, and my beloved sister, Laís, for the infinite, implacable, and unconditional love, for the example of perseverance, for all support and dedication, and for being my safe harbor in times of storm.

I would like to thank the awesome members of the Laboratory of Information Systems (LIS), for the moments of learning, laughing, support, more laughing, and true friendship.

I would like to thank Professor Shazia Sadiq, for the opportunities; to Sandra Goutte, Professor Julio Cesar, and Jaudete Daltio for the crucial contributions.

I would like to thank my huge, crazy, and amazing family, for the immeasurable love.

I would like to thank my beloved Livia, for the unconditional support and love; to my dearest friends Ivo, Augusto, Dudu, Ricardo, Rafael, Paola, Graça, and the 247 squad, for being there for me and for putting a smile in my face no matter what.

I would like to thank colleagues, faculty, and staff of the Institute of Computing and of UNICAMP, for all support and companionship that created a healthy environment for me and for the development of this work.

I would like to thank Xastre, Peternela, and every outstanding member of the Supê crew, for the unsurpassed patience and support.

I would like to especially thank so many extraordinary people that believed, supported, and collaborated with me and this work in so many levels, but unfairly remain anonymous in these acknowledgments. To all of you, my deepest, sincere, and eternal gratitude.

# Resumo

As ciências biológicas e médicas precisam cada vez mais de abordagens unificadas para a análise de dados, permitindo a exploração da rede de relacionamentos e interações entre elementos. No entanto, dados essenciais estão frequentemente espalhados por um conjunto cada vez maior de fontes com múltiplos níveis de heterogeneidade entre si, tornando a integração cada vez mais complexa. Abordagens de integração existentes geralmente adotam estratégias especializadas e custosas, exigindo a produção de soluções monolíticas para lidar com formatos e esquemas específicos. Para resolver questões de complexidade, essas abordagens adotam soluções pontuais que combinam ferramentas e algoritmos, exigindo adaptações manuais. Abordagens não sistemáticas dificultam a reutilização de tarefas comuns e resultados intermediários, mesmo que esses possam ser úteis em análises futuras. Além disso, é difícil o rastreamento de transformações e demais informações de proveniência, que costumam ser negligenciadas.

Este trabalho propõe LinkedScales, um dataspace baseado em múltiplos níveis, projetado para suportar a construção progressiva de visões unificadas de fontes heterogêneas. LinkedScales sistematiza as múltiplas etapas de integração em escalas, partindo de representações brutas (escalas mais baixas), indo gradualmente para estruturas semelhantes a ontologias (escalas mais altas). LinkedScales define um modelo de dados e um processo de integração sistemático e sob demanda, através de transformações em um banco de dados de grafos. Resultados intermediários são encapsulados em escalas reutilizáveis e transformações entre escalas são rastreadas em um grafo de proveniência ortogonal, que conecta objetos entre escalas. Posteriormente, consultas ao dataspace podem considerar objetos nas escalas e o grafo de proveniência ortogonal. Aplicações práticas de LinkedScales são tratadas através de dois estudos de caso, um no domínio da biologia – abordando um cenário de análise centrada em organismos – e outro no domínio médico – com foco em dados de medicina baseada em evidências.

# Abstract

Biological and medical sciences increasingly need a unified view of multiple data sources for exploring the network of relationships and interactions among elements. Nevertheless, the construction of such network has been increasing in complexity as essential data are frequently scattered across an ever-growing set of sources with multiple levels of heterogeneity. Existing data integration approaches usually adopt specialized, heavyweight strategies, requiring a costly upfront effort to produce monolithic solutions for handling specific formats and schemas. These unsystematic approaches produce ad-hoc solutions, hampering the reuse of partial integration tasks and intermediary outcomes. Furthermore, provenance information useful for future analysis is usually neglected.

This work proposes LinkedScales, a multiscale-based dataspace designed to support the progressive construction of a unified view of heterogeneous data sources. It systematizes complex integration chains and encapsulates intermediary outcomes as scales, departing from raw representations (lower scales), incrementally going towards ontology-like structures (higher scales). LinkedScales defines a data model and a systematic, on-demand integration process via transformations over a graph database. Inter-scale transformations are tracked in an orthogonal provenance graph connecting objects between scales. Queries over LinkedScales can consider both the scales and the orthogonal provenance graph. Practical applications of LinkedScales are discussed through two case studies, one on the biology domain – addressing an organism-centric analysis scenario – and another on the medical domain – focusing on evidence-based medicine data.

# List of Figures

2.1	Profile integrating characteristics scattered across several sources . . . . .	19
2.2	Overview of the <i>LinkedScales Primary Data Architecture</i> . . . . .	21
2.3	<i>LinkedScales</i> graph model . . . . .	23
2.4	Example of a match/transform process . . . . .	25
2.5	Example of transformation between two scales and the corresponding MTG	26
2.6	Excerpt of a XLS spreadsheet highlighting the row regarding the species <i>Brachycephalus ephippium</i> . . . . .	28
2.7	Excerpt of a XML/NEXUS file highlighting the species <i>Brachycephalus</i> <i>ephippium</i> . . . . .	29
2.8	Graph Representation of an XLS file as a graph in the Physical Scale . . .	29
2.9	Graph Representation of an XML/NEXUS file as a graph in the Physical Scale . . . . .	30
2.10	All stages presented as a graph-based representation . . . . .	31
2.11	Example of visualization of the Description Scale . . . . .	32
3.1	Profile integrating characteristics scattered across several sources . . . . .	35
3.2	<i>LinkedScales Primary Data Architecture</i> [56] . . . . .	37
3.3	Schematic illustration of the bootstrap phase of the experiment . . . . .	38
3.4	Excerpt of a XLS and its representation on the <i>Physical</i> and <i>Logical scales</i>	39
3.5	Logical-description trails driving a logical-description scale transformation, and description-conceptual trails driving the production of the conceptual scale . . . . .	40
3.6	Comparison of the same portion of two conceptual scales: Non trail-based (left) and trail-based (right) transformations . . . . .	42
4.1	For patients presenting with acute chest pain, the clinical features and the respective Likelihood Ratio concerning the probability of a Acute Myocar- dial Infarction (source [64]). . . . .	45
4.2	Nomogram applying the LR of 7.1 over the probability of 25%. . . . .	46
4.3	Relationships among clinical features and diagnosis compiled in a graph- like structure plus the case planned for the game (black nodes) and the route pursued by the player (white nodes with two borders). . . . .	47
4.4	CASNET 3D Description of Glaucoma . . . . .	49
4.5	LinkedScales Primary Data Architecture applied to the EBM Scenario. . .	52
4.6	Example of a path and object within a scale. . . . .	55
4.7	Example of transformation: two objects as input, a third object as output	57
4.8	Connecting elements from Heart Failure Ontology with data extracted from a meta-analysis paper (table from [44]) . . . . .	58
4.9	Overview of the implemented LinkedScales architecture . . . . .	59

4.10	Example showing the sequence of transformations from the Physical to the Conceptual scale . . . . .	60
4.11	Example of a criterion transforming data from the description to the conceptual scale . . . . .	61
4.12	Example of Clinical Findings and related concepts found by the OntoMatch service. . . . .	62
4.13	An example illustrating an association of a clinical evidence to an entity and the respective MTG. . . . .	63
4.14	Relation between the average number of candidate concepts per Clinical Finding and the minimum similarity threshold. . . . .	64
4.15	Filtering and annotating evidences according to similarity. . . . .	65
4.16	Screenshot of the prototype for navigating in the conceptual scale. . . . .	66
A.1	Overview of the LinkedScales architecture. . . . .	81
A.2	Overview of the current state of the IMM. Source: <a href="http://www.omgwiki.org/imm">www.omgwiki.org/imm</a> . . . . .	82
A.3	Main idea behind the work [59]: A PDF document and its corresponding shadow. . . . .	83
A.4	Shadows approach presented in a LinkedScales perspective. . . . .	84
A.5	Linked Biology project presented in a LinkedScales perspective. . . . .	85
A.6	Spreadsheet integration presented in a LinkedScales perspective. . . . .	86
A.7	Spreadsheet data articulation via entity recognition. . . . .	86
A.8	Screencopy of our prototype integrating data of several spreadsheets. . . . .	87

# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Motivation . . . . .	13
1.2	Overview of the proposal and contributions . . . . .	14
1.3	Thesis organization . . . . .	15
<b>2</b>	<b>Multiscaling a Graph-based Dataspace</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Foundations and Related Work . . . . .	18
2.2.1	Challenges on organism-centric analysis for data integration . . . . .	18
2.2.2	Upfront Data Integration <i>vs.</i> The “Pay-as-you-go” Integration . . . . .	19
2.3	LinkedScales Framework . . . . .	20
2.4	Multiscale Graph Model . . . . .	22
2.4.1	Preliminary Definitions . . . . .	23
2.4.2	Transformation process . . . . .	24
2.4.3	Multiscale Transformations and the Transformation Graph . . . . .	26
2.5	Experimental Scenario: Organism-Centric Analysis via LinkedScales . . . . .	27
2.5.1	Implementing the Solution . . . . .	27
2.5.2	Scenario and Experimental Procedure . . . . .	28
2.5.3	Ingestion: From the original sources to the physical scale . . . . .	28
2.5.4	From the Physical to the Logical scale . . . . .	30
2.5.5	From the Logical to the Description scale . . . . .	30
2.5.6	From the Description to the Conceptual scale . . . . .	32
2.6	Conclusion . . . . .	32
<b>3</b>	<b>Data Integration and Semantic Enrichment</b>	<b>34</b>
3.1	Introduction . . . . .	34
3.2	Foundations and Related Work . . . . .	35
3.3	LinkedScales . . . . .	36
3.4	Combining Trails with LinkedScales . . . . .	38
3.4.1	Physical-logical Trails . . . . .	39
3.4.2	Logical-description Trails . . . . .	40
3.4.3	Description-conceptual Trails . . . . .	41
3.5	Conclusion . . . . .	42
<b>4</b>	<b>Multiscaling a Finding-Disease Dataspace</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	EBM Data as a Basis for Medical Training . . . . .	44
4.2.1	EBM Knowledge Base . . . . .	44
4.2.2	Game and Knowledge Base Interaction . . . . .	46

4.3	Related Work . . . . .	48
4.3.1	Computer-based Medical Learning . . . . .	48
4.3.2	EBM and Consensual Knowledge . . . . .	49
4.3.3	Classical vs. Progressive, On-demand Data Integration . . . . .	50
4.4	LinkedScales . . . . .	50
4.4.1	The Data Integration Architecture . . . . .	51
4.4.2	The underlying multiscale graph-based data model . . . . .	53
4.4.3	Extending the data model . . . . .	54
4.5	Building a Clinical Feature-Disease Dataspace . . . . .	57
4.5.1	Experimental Scenario . . . . .	58
4.5.2	Implementation Aspects . . . . .	59
4.5.3	From sources to the Conceptual Scale . . . . .	60
4.5.4	Entity Resolution in the Conceptual Scale . . . . .	62
4.5.5	Multiscale Transformation Graph and Uncertainty . . . . .	63
4.5.6	Navigating on the Conceptual Scale . . . . .	66
4.6	Conclusion . . . . .	67
<b>5</b>	<b>Conclusions and Future Work</b>	<b>68</b>
	<b>Bibliography</b>	<b>70</b>
<b>A</b>	<b>Conceiving a Multiscale Dataspace for Data Analysis</b>	<b>78</b>
A.1	Introduction and Motivation . . . . .	78
A.2	Related Work . . . . .	79
A.2.1	The Classical Data Integration . . . . .	79
A.2.2	The “Pay-as-you-go” Dataspace Vision . . . . .	80
A.3	LinkedScales: A Multiscale Dataspace Architecture . . . . .	80
A.4	Previous Work . . . . .	83
A.4.1	Homogeneous Model – Universal Lens for Textual Document Formats	83
A.4.2	Connecting descriptive XML data – a Linked Biology perspective .	84
A.4.3	Progressively Integrating Biology Spreadsheet Data . . . . .	85
A.5	Concluding Remarks . . . . .	87



# Chapter 1

## Introduction

### 1.1 Motivation

A consequence of the intensive growth of information shared online is the increase of opportunities that emerge from the exploitation of links across distinct sources of knowledge [7, 37, 45]. Biology and health are two domains that can be highly benefited by this exploitation and are a focus of this work [86, 11].

Biology is a domain increasingly exploring unified views of diverse resources to understand and discover relationships between low-level (e.g., cellular, genomic or molecular level) and high-level (e.g., species characterization, macrobiomas, etc.) [62]. However, the construction of such network-like view is hampered by different levels of heterogeneity in available resources [31, 10].

In the health domain, the EBM (Evidence-Based Medicine) is an information-driven field, which demands efforts to put together and link data from different sources [11]. Aiming at increasing the use of conscientious and rational clinical decision making, EBM proposes the use of evidence reported by reliable and well-conducted research [73, 70]. Data concerning evidence-based medicine is scattered across multiple publications and their associated outcomes (papers, spreadsheets, etc.), being usually segmented according to illness. Such scenario hinders, for instance, the exploration of a cross-connected view of symptoms and diseases reported in the available publications [11].

To integrate available sources, classical *heavyweight* data integration approaches usually require costly upfront efforts to handle specific formats and schema recognition/mapping tasks, frequently resulting in ad-hoc, monolithic solutions [75, 50, 33]. Furthermore, such unsystematic solutions do not foster the reuse of intermediary integration outcomes and frequently neglect provenance information [29, 34].

As a modern alternative to the *all-or-nothing* integration solutions, *Franklin et al.* [26] propose the notion of *dataspaces*. They argue that linking lots of “fine-grained” information particles, bearing “little semantics”, already bring benefits to applications, and more links can be produced on demand, as *lightweight* steps of integration.

Related work proposals address distinct aspects of dataspaces. They include approaches of incremental integration based on user-feedback [40, 6, 5]; techniques for querying and indexing dataspace [87, 22]; and model mappings for representing different types of sources within a dataspace [35, 20, 85]. Furthermore, proposed Dataspace Support Plat-

forms address a variety of specific scenarios, e.g., SEMEX [14] and iMeMex [19] on the Private Information Management context; PayGo [53] focusing on Web-related sources; and a solution for justice-related data [77]. However, no dominant proposal of a complete architecture has emerged to date [75, 34, 32].

Based on architectural aspects investigated in available dataspace solutions and on several previous experiences, we observed the lack of a systematic approach to address the different steps involved in an integration using dataspaces. Therefore, we propose here a dataspace framework that provides such systematization via the notion of progressive scales.

## 1.2 Overview of the proposal and contributions

This thesis proposes a dataspace framework, called *LinkedScales*, capable of distinguishing integration steps accordingly to interdependent roles, slicing and organizing the process as a stack of abstraction layers (scales), each scale having specialized algorithms and services.

*LinkedScales* defines an architecture built over an instance of the proposed multiscale graph-based data model. Inspired by the encapsulation principle of multilayered software architectures, each scale hides from its upper scale the specificities of the data it receives as input, presenting a standard interface according to its role. It enables to factor the different aspects of the problem per scale and to reuse scale-specialized algorithms.

LinkedScales takes advantage of the flexibility of graph structures and proposes the notion of scales of integration. Scales are represented as graphs, managed in graph databases. Operations become transformations of such graphs. LinkedScales also systematically defines a set of scales as layers, where each scale focuses on a different level of integration and its respective abstraction. Scale transformations within the dataspace are tracked by an orthogonal graph, supporting traceability among tasks through the correlation of sources and targets to transformations between scales. Furthermore, LinkedScales supports a complete dataspace lifecycle, including automatic initialization, maintenance, and refinement of the links.

The proposed on-demand refinement strategy is inspired by the approach envisaged by [78], which is directed by trails (scale-specialized annotations). Details regarding the proposal and its application in two case studies are presented in next chapters.

The four main contributions of this work are (i) the definition of a **Multiscale Data Model** based on graphs, which enables to organize the dataspace in scales interrelated by an orthogonal graph; (ii) the definition of an **Integration Architecture**, based on previous work, that systematizes steps of integration as scales; the adoption of a (iii) **Dataspace Refinement Strategy** based on scale-specialized annotations (trails); and the (iv) **implementation and evaluation of the proposal** by adopting it as the underlying basis for integrating biological and health data.

The three fundamental benefits of LinkedScales are (i) the systematization of integration steps; (ii) the record of provenance between integration steps; and (iii) the support for reuse of partial outcomes.

### 1.3 Thesis organization

This chapter presented a brief overview of the context, faced challenges, the proposed solution, and the contributions of this research. The reminder chapters are organized as a collection of papers, as follows.

Chapter 2 corresponds to the paper "*Multiscaling a Graph-based Dataspace*", published in the *Journal of Information and Data Management* [56]. The article introduces the proposed multiscale-based dataspace architecture and defines the graph-based data model. Furthermore, the paper presents initial implementations of the proposal applied to the biology domain – addressing the organism-centric analysis scenario.

Chapter 3 corresponds to the paper "*Progressive Data Integration and Semantic Enrichment Based on LinkedScales and Trails*", published in the *Proceedings of the 9th International Conference on Semantic Web Applications and Tools for Life Sciences* [60]. This paper extends previous publications by introducing *trails*, lightweight, scale-specialized semantic annotations for dataspace refinement. Trails are used to support a progressive building of semantic representations.

Chapter 4 corresponds to the paper "*Multiscaling a Finding-Disease Dataspace*". This paper, currently under review, extends the previous definition of the data model to add the fundamental concept of objects to support the tracking mechanism. The paper also presents the implementation of our architecture in the construction of an evidence-based medicine network.

Chapter 5 presents conclusions and future directions for the work.

Appendix A corresponds to the paper "*Conceiving a multiscale dataspace for data analysis*". This paper presents an historic overview of the development of our architecture, discussing how previous experiences guided its model. It is attached to this thesis as a complementary resource.

Besides the publications in Chapters 2 and 3, other papers associated with this research were published in the course of this thesis, listed as follows.

- Matheus Silva Mota, Julio Cesar dos Reis, Sandra Goutte, and André Santanchè. Multiscale dataspace for organism-centric analysis. Proceedings of the XXX Brazilian Symposium on Databases (SBBD). 2015.<sup>1</sup> [57]
- Matheus Silva Mota and André Santanchè. *Conceiving a multiscale dataspace for data analysis*. Proceedings of the Brazilian Conference on Ontologies (Ontobras). 2015. [61]
- Matheus Silva Mota, Julio Cesar dos Reis, Sandra Goutte, and André Santanchè. *Multiscaling a graph-based dataspace*. Journal of Information and Data Management (JIDM), page 16, 2016. [56]
- Matheus Silva Mota, Fagner Leal Pantoja, Júlio Cesar dos Reis, and André Santanchè. *Progressive data integration and semantic enrichment based on linked scales and trails*. Proceedings of the 9th International Conference Semantic Web Applications and Tools for Life Sciences (swat4ls). 2016. [60]

---

<sup>1</sup>Received the best short paper award

- Matheus Silva Mota, Francisco José Nardi Filho, Roger Vieira Horvat, Marcelo Schweller, Tiago de Araujo Guerra Grangeia, Marco Antonio de Carvalho Filho, Júlio Cesar dos Reis, Fagner Leal Pantoja, André Santanchè. *Multiscaling a Finding-Disease Dataspace*. (under review)

## Chapter 2

# Multiscaling a Graph-based Dataspace

### 2.1 Introduction

Data-centric domains as biology are increasingly adopting different systems to produce, store and analyze datasets regarding specific processes and aspects of biological organisms – *e.g.*, experiments, descriptions, collections, simulations, *etc.* However, heterogeneity hampers the integrated exploration of knowledge across systems and research groups [37]. Therefore, integration remains a key issue since providing a “big picture” view of data may offer new perspectives and insights for researchers [24, 67].

This research focuses on a specific data integration paradigm known as Dataspaces [26]. It advocates the advantages of an on-demand lightweight integration to comply with the dynamicity of modern environments, against the classic heavyweight upfront techniques. One of the advantages of on-demand integration is the ability of readily shaping the final outcome according to present needs. A key issue with on-demand integration, addressed in this article, refers to the long chain of steps from source to target. In one extreme, biologists want to treat knowledge at a conceptual level, handling data in an integrated fashion. In the other extreme, there are several problem-relevant heterogeneous data sources, comprising files, DBs, ontologies, *etc.* Between both extremes, there might have a spectrum of intermediary integration steps, which are difficult to determine.

In this article, we propose an approach named *LinkedScales*, which aims at splitting the integration steps as discrete scales. Each scale encompasses common aspects and routines related to a specific integration step. The main objective of *LinkedScales* is to go from a source-related lower scale to a user-focused higher scale. Inspired by the layered software architecture, each scale offers to the immediate upper scale a pre-agreed model (interface), encapsulating a given type of heterogeneity of the lower scale. This investigation defines the different scales, formalizing them in a framework based on a graph model. In lower scales, we depart from a myriad of heterogeneous sources available. The upper scales enables to tailor the model according to specific needs, *i.e.*, the integration model fits the user needs, instead of the opposite.

We demonstrate the applicability of our proposal in the biological domain. In such dynamic context, reuse plays a key role and traditional on-demand solutions usually rely on ad-hoc techniques, implementing the entire integration chain. In our proposal, the encapsulation of scales in *LinkedScales* enables to customize only algorithms of a

specific scale, reusing the remaining of the chain. Obtained results relying on real-world application scenarios experimenting the approach indicate the adequacy and usefulness of the *LinkedScales* proposal for organism-centric analysis.

The remaining of this article is organized as follows: Section 2.2 presents the problem in our research scenario and how existing work concerning data integration address it. Section 2.3 reports on the proposed *Linkedscales* framework. Section 2.4 details the formalization of the multiscale graph model. Section 2.5 describes implementation aspects and experiments showing a complete example to illustrate the proposed approach. We also discuss its benefits. Finally, Section 2.6 wraps up the article with conclusions and presents future work.

## 2.2 Foundations and Related Work

### 2.2.1 Challenges on organism-centric analysis for data integration

Organism-centric analysis refers to an usual approach conducted by biologists in which organisms – *i.e.*, species or taxonomic groups – are the central focus of the analysis and data are integrated around them. A common task faced by biologists conducting an organism-centric research refers to the construction of "views" of data, we call here *profiles* [81]. Profiles vary according to the focus of interest, but they can be seen as a subset of descriptive data of organisms selected for a research [36]. The construction of such profiles involves combining data usually fragmented in heterogeneous sources, requiring further efforts from biologists to collect and combine pieces coming from multiple repositories and several files with different formats.

Consider the example of profile illustrated in Figure 2.1, defined by biologists interested in validating hypotheses regarding the evolution of “deafness” in frogs. Aiming at understanding why distant phylogenetic groups of frogs lack middle ear structures, biologists want to gather together as profiles data regarding morphological traits, habitat, reproduction mode, acoustics and phylogenetic trees of several species. Morpho-anatomical data would be required to examine whether miniaturisation in frogs lead to the loss of ear structures, while acoustic data would allow testing the co-evolution of mutism and deafness, *etc.* Based on such profiles, biologists might compare organisms in a systematic way and investigate conditions and associations related with the hypotheses.

Phylogenetic data for the target species of the genus *Brachycephalus* (shown in the center area of Figure 2.1) can be found within the *TreeBASE*<sup>1</sup> repository – where scientists share their experimental data files – as a XML/Nexus file. It contains the phylogenetic tree reconstructed from DNA sequences from a study. Records from IUCN Red List<sup>2</sup> intended for conservation contains data regarding the species habitat in CSV format. Moreover, several phenotypic data can be found from Quaaardvark System<sup>3</sup> in Excel format.

In this scenario (Figure 2.1), biologists spend a lot of time “cutting and pasting” data from each of the sources and organizing them in spreadsheets before any analysis. On the other hand, a systematic integration approach requires several steps of integration,

---

<sup>1</sup><http://treebase.org>, <sup>2</sup><http://www.iucnredlist.org>, <sup>3</sup><http://animaldiversity.ummz.umich.edu/quaaardvark>

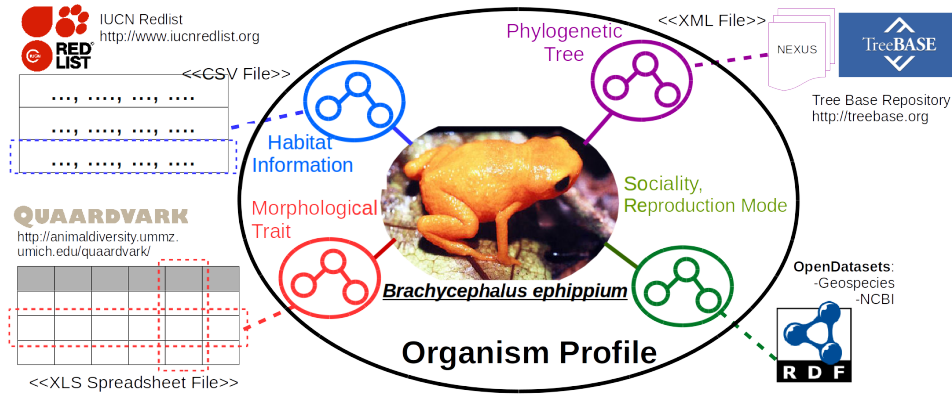


Figure 2.1: Profile integrating characteristics scattered across several sources

due to the different types of heterogeneity, *i.e.*, different formats (CSV, Excel, Nexus), different structures (tables, trees), different schemas, *etc.* Therefore, the combination of different types of datasets may prove challenging, and the integration of missing data often result in a drastic data trimming and the partial use of the data available. Furthermore, such biological research has an intrinsic dynamism. For instance, biologists may discover during their investigations that other characteristics must be taken into account, which might require further efforts to reflect the new requirements and data on the profiles to make them up-to-date.

### 2.2.2 Upfront Data Integration *vs.* The “Pay-as-you-go” Integration

Motivated by the increasingly need of treating multiple and heterogeneous data sources, data integration has been the focus of attention in the database community in the past two decades [35, 32]. Several data integration strategies have emerged, including federated databases, schema integration and data warehouses [28, 69, 46, 30].

A common adopted approach relies on providing a virtual unified view under a global schema (GS) [75, 46]. Within GS-based systems, the data stay in their original data sources – *i.e.* maintaining their original schemas – and are dynamically fetched and mapped to a global schema under clients’ request [50, 35]. In a nutshell, applications send queries to a mediator, which relates them into several sub-queries dispatched to wrappers, according to meta-data regarding capabilities of the participating database management systems (DBMSs). Wrappers map queries to the underlying DBMSs and the results back to the mediator, guided by the global schema. Queries are optimized and evaluated according to each DBMS within the set, providing the illusion of a single database to applications [50].

The central drawback with such data integration strategy regards the big upfront effort required to produce a global schema definition [30]. As in some domains different DBMSs may emerge and schemas are constantly changing, such costly initial step can become impracticable [35]. Moreover, several approaches focus on a particular data model (*e.g.*, relational), while new models also become popular [23]. As proposed in this investigation,

our approach supports progressive small integration steps as an alternative to this classical all-or-nothing costly upfront data integration technique.

Since upfront mapping between schemas are labor intensive and scheme-static domains are rare, pay-as-you-go integration strategies have gained momentum. Classical data integration approaches might work successfully when integrating modest numbers of stable databases in controlled environments. Nevertheless, literature still lacks an efficient and definitive solution for scenarios in which schemas often change and new data models must be considered [35]. In a data integration spectrum, the classical data integration is at the high-cost/high-quality end, while an incremental integration based on progressive small steps starts in the opposite side. Such incremental integration can be continuously refined in order to improve the connections among sources.

The notion of *dataspaces* aims at providing the benefits of the classical data integration approach, but in a progressive fashion way [29, 75, 34]. The main argument behind the dataspaces proposal is that, in the current scenario, instead of a long wait for a global integration schema to have access to the data, users would rather to have early access to the data, among small cycles of integration – *i.e.*, if the user needs the data now, some integration is better than nothing.

Dataspaces approach of data integration can be divided in a bootstrapping stage and subsequent refinements. Progressive integration refinements can be based, for instance, on structural analysis [22], on users’ feedback [6] or on manual/automatic mappings among sources – if benefits worth such effort. Furthermore, several Dataspace platforms address a variety of specific scenarios, *e.g.*, SEMEX [14] and iMeMex [19] on the private information management context; PayGo [53] focusing on Web-related sources; and a justice-related dataspace [77].

Although incremental integration approaches have already showed their potentialities, literature still lacks an architecture that systematizes the progressive integration steps and results according to integration aspects, providing provenance and reuse of partial results. Systematization, provenance and reuse are the three pillars of our *LinkedScales* proposal, introduced in next section.

## 2.3 LinkedScales Framework

*LinkedScales* refers to a framework that comprises a multiscale graph model – introduced here and formally detailed in the next section – and a data architecture which instantiate the model. It aims at bringing the proposal of multiscale to the data integration chain, systematizing and encapsulating the data regarding integration steps as graph-based scales. In our approach, the modern tendency towards progressive integration [29] evolves in progressive steps within a shared “*space*”, in which data of several steps coexist, even if not fully integrated. Over time, extra incremental integration steps are made within the space when benefits worth the efforts.

*LinkedScales* is based on an abstract model that organizes the progressive integration chain as a pile of scales, where the entities in an upper scale are built based on transformations over entities of a lower scale – the granularity and semantics of the entities vary



according to the scale. The integration starts on the lowest scale, where all original data sources are ingested and transformed into graphs. Each subsequent scale from this point is a graph derived from the previous scale, taking advantage of the flexibility of graphs to logically represent different structures along the scales. This model allows representing operations within and across the scales as transformation procedures in graphs. Scales are interconnected by an orthogonal graph, supporting traceability among them – *i.e.*, it is possible to "track" sources/targets of transformations between scales.

In order to address a range of applications which share common data integration concerns, we propose a *LinkedScales Primary Data Architecture*, defining a starting set of scales, based on previous experiences on data integration [59, 8, 55]. Each scale of this data architecture emphasizes a different level of integration and its respective abstraction.

Figure 2.2 presents an overview of the *LinkedScales Primary Data Architecture*. It depicts four different scales of abstraction aiming at going from the raw data sources (lower scales, containing more details about format and structure) to a conceptual scale (fewer details of format and structure, and focus on domain-specific concepts). From bottom to top, the scales are: (i) *Physical Scale*, (ii) *Logical Scale*, (iii) *Description Scale*, and (iv) *Conceptual Scale*. This primary data architecture was conceived to be extended, *i.e.*, further scales can appear on top of the conceptual scale to define additional domain-related views.

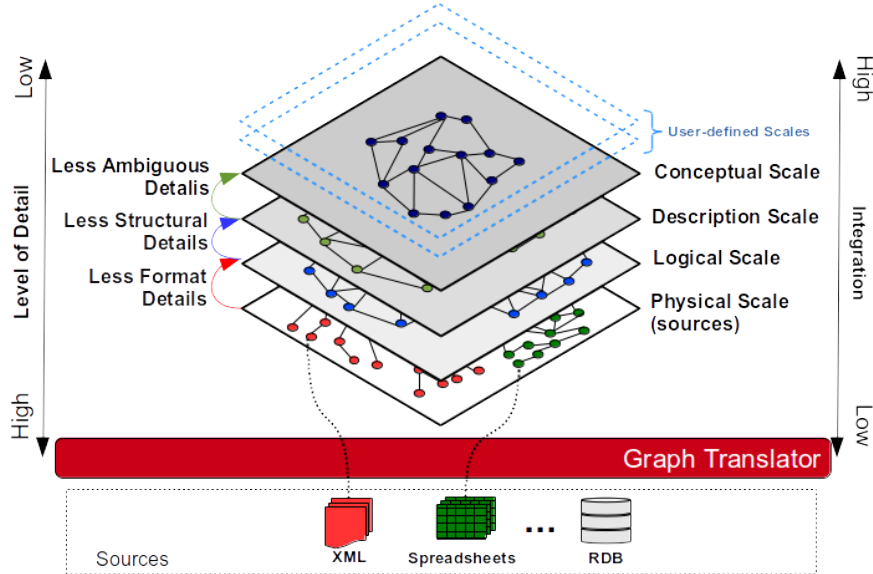


Figure 2.2: Overview of the *LinkedScales Primary Data Architecture*

The lowest scale in Figure 2.2 – **Physical Scale** – aims at representing the different data sources in their original physical format as a graph. The original raw data sources are transformed into a graph by an ingestion procedure (the Graph Translator in the figure) able to read several specialized formats – *e.g.*, Excel, CSV, relational tables, XML – and convert them to an equivalent graph representation. The original structure, format and content of the underlying data sources are reflected in a graph as far as possible. The role of this scale is to homogenize the physical representation, making explicit and linkable elements of the original data within sources.

Based on experiences of a previous work that explores a homogeneous representation model for textual documents independently of formats [59], the next scale proposed is the ***Logical Scale***. It offers a common view to data inside similar or equivalent structural models represented in the previous scale. Tables and hierarchical documents are examples of structural models present in the sources containing data regarding organisms. In the previous scale, differences might exist in the representation of a table within a PDF, a table from a spreadsheet and a table within a HTML file, since they preserve specificities of their formats. Within the *Logical Scale*, format specificities disappear and the three tables are represented alike since they refer to the same structural model. This leads to a homogeneous approach to process tables, independently of the way that tables are represented in their original specialized formats.

The ***Description Scale*** emphasizes the content (*e.g.*, labels of elements within an XML document or values in spreadsheet cells) and their relationships. Since models represent relations among data elements in different ways – *e.g.*, a row in a table can represent data concerning the same entity while hierarchical relations in a document represent aggregations – the *Description Scale* reduces all logical models to a single unified one, to shift the focus towards the descriptive content, avoiding heterogeneous models concerns.

The unified model selected for this scale relies on the triple  $\langle \text{resource}, \text{property}, \text{value} \rangle$ , which is usual in several meta-data standards as *Resource Description Framework* (RDF<sup>2</sup>). This scale only unifies the logical model, but still lacks essential properties of a semantic representation like RDF since it does not: distinguish entities, adopt controlled vocabularies to represent descriptive properties or make explicit the semantics of the elements using ontologies. This stands for the role of the next scale.

The highest scale of our data architecture, illustrated in Figure 2.2, refers to the ***Conceptual Scale***. It integrates data of the lower scale in a semantic level, exploits the content and relationships between nodes to discover and to make explicit through ontologies their latent semantics. Entities are discovered, deduplicated and related to ontologies as instances of classes, or properties and their values. Therefore, a “textual graph” of the previous scale becomes a graph containing interrelated entities and their properties/values, with explicit semantics supported by ontologies. We also consider that predefined ontologies can be straightly interrelated to this scale, to be linked to the inferred entities.

## 2.4 Multiscale Graph Model

This section adopts a formal language to define aspects of the abstract model underlying the *LinkedScales* approach introducing our *Multiscale Graph Model*. It aims at facilitating the understanding of the involved concepts, but, it is not a full-fledged formal definition of the model. We organize three subsections, presenting first the preliminary definitions, followed by the transformation process and the orthogonal transformation graph.

---

<sup>2</sup><https://www.w3.org/RDF/>

### 2.4.1 Preliminary Definitions

As depicted in Figure 2.3, the *Multiscale Graph Model* contains a sequence of scales  $(S_1, S_2, \dots, S_n)$ . It starts from an initial scale  $S_1$  and each subsequent scale  $S_i$  is derived from a previous scale  $S_{i-1}$ .

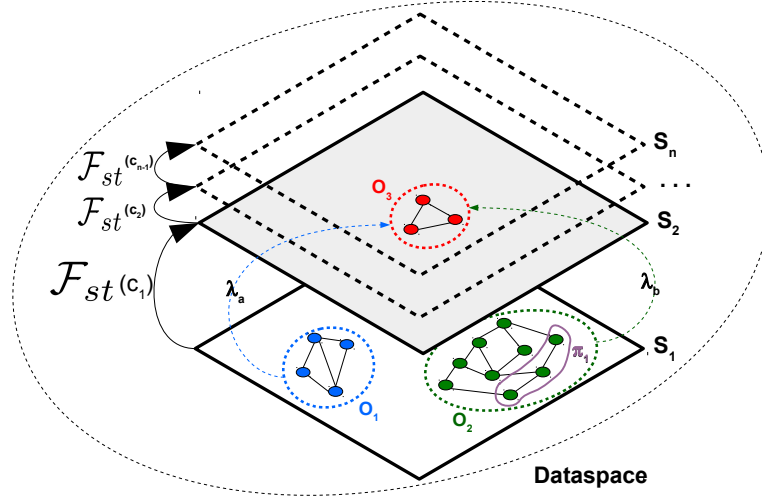


Figure 2.3: *LinkedScales* graph model

Inspired by the notion of *graph databases*, a **Scale** is defined as a finite, edge-labeled, directed graph [83, 4, 3, 17]. Formally, let  $\Sigma$  be a finite alphabet and  $\mathcal{V}$  be a countably infinite set of node ids. A scale  $S$  over  $\Sigma$  is a pair  $(V, E)$ , being  $V$  a finite set of **nodes** and  $E$  a finite set of **edges**, where  $V \subseteq \mathcal{V}$  and  $E \subseteq V \times \Sigma \times V$ . Furthermore, given any two scales  $S_i = (V_i, E_i)$  and  $S_j = (V_j, E_j)$ , where  $V_i \subseteq \mathcal{V}$  and  $V_j \subseteq \mathcal{V}$ ,  $V_i \cap V_j = \emptyset$ .

Given a scale  $S = (V, E)$  and two nodes  $u, v \in V$  and a label  $a \in \Sigma$ , an edge  $e \in E$  is a triple  $(u, a, v)$  indicating a link between  $u$  and  $v$  with a label  $a$ . A **path**  $\pi$  in a scale  $S$  is a set of edges in  $E$  connecting two nodes (initial and final) in  $V$ . Therefore, a path connecting a node  $v_1$  and  $v_m$  is a sequence of edges  $\pi = \{(v_1, a_1, v_2), (v_2, a_2, v_3), \dots, (v_{m-1}, a_{m-1}, v_m)\}$ , where any edge  $(v_{i-1}, a_{i-1}, v_i) \in E$  and any end node of an edge in the path matches the initial node in the following edge. An empty path  $\pi$  is a triple  $(v, \epsilon, v)$ , where  $v \in V$  and the label is the empty word  $\epsilon$ ; the length of such path,  $|\pi| = 0$ . The concept of path plays a key role in our transformation process.

A transformation between two scales is defined in terms of transformations of objects inside these scales, *i.e.*, objects are the atomic transformation units. An **object** is defined as a set of paths  $O = \{\pi_1, \pi_2, \dots, \pi_r\}$ . An object  $O_h$  belongs to a scale  $S_i$  if all nodes/edges of the paths in  $O_h$  are nodes/edges of  $S_i$ . Figure 2.3 depicts three objects and a path,  $O_1$ ,  $O_2$  belongs to  $S_1$ ,  $O_3$  belongs to  $S_2$ , and the path  $\pi_1 \in O_2$ .

### 2.4.2 Transformation process

*LinkedScales* is represented as tuple  $\mathcal{LS} = (S_i, \Omega, \mathcal{F}_{st})$ , where  $S_i$  is a scale representing the initial state,  $\Omega = \{C_1, C_2, \dots, C_n\}$  is a sequence of transformation criteria and  $\mathcal{F}_{st}$  is a function  $\mathcal{F}_{st} : S_i \rightarrow S_{i+1}$  which derives a subsequent scale  $S_{i+1}$  by applying a transformation criteria  $C_i$  over a previous scale  $S_i$ .

The **transformation** process comprises two steps: match and transform. The *match* step aims at finding paths in the subgraphs of a given scale, while the *transform* step addresses the production of a transformed subgraph in the upper scale. The example illustrated in Figure 2.5 shows how an instance of a table ( $T_1$ ) with a schema and two rows results in two entities ( $e_4$  and  $e_5$ ), each one containing three *paths* representing RDF-like triples. Such transformation is based on a pattern for matching paths in the input and for creating the corresponding nodes and vertices in the output.

---

**Algorithm 1** Scale Transformation

---

```

1: procedure  $\mathcal{F}_{st}(S_i, C_i)$      $\triangleright$  Produces a scale  $S_{i+1}$  based on a scale  $S_i$  and criteria  $C_i$ 
2:    $S_{i+1} \leftarrow \emptyset$ 
3:   for  $\lambda \in C_i$  do
4:      $\mathcal{O} \leftarrow \lambda_{match}(S_i)$                                  $\triangleright$  Returns all matched objects in  $S_i$ 
5:     for object  $O \in \mathcal{O}$  do
6:        $S_{temp} \leftarrow \lambda_{transform}(O)$ 
7:        $S_{i+1} \leftarrow (S_{i+1} \cup S_{temp})$ 
8:     end for
9:   end for
10:  return  $S_{i+1}$ 
11: end procedure

```

---

The match and transform operations are encapsulated in the concept of criterion. A **criterion**  $\mathcal{C}_\alpha$  is a set of criterion  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ . A **criterion**  $\lambda_a$  is a pair  $(m_a, t_a)$ , where  $m_a$  is a *match* operation and  $t_a$  is a *transform* operation. Similarly to the *SELECT* operator in SQL, a **match** operator meets a set of objects of a given scale  $S_i$ , while the respective **transform** operator derives these objects to produce a graph in  $S_{i+1}$ . Algorithm 1 describes how the *scale transformation* function produces an upper scale based on a criteria set.

A pattern in the *match* operation is defined by a regular expression over the graph. Wildcards here are indicating the repetition of sub-patterns. While the wildcard *\** indicates a repetition of a given subpath in a sequential disposition – *i.e.*, the beginning of a repeated subpath is connected with the end of the previous one – the wildcard *\*\** indicates a repetition in a parallel disposition – *i.e.*, all repeated subpaths are connected to the same origin. The number of repetitions is constrained by adding the clause  $[\alpha..\beta]$ , where  $\alpha$  and  $\beta$  are optionals minimum and maximum boundaries, respectively.

Figure 2.4 visually illustrates the steps in a transformation that aims at producing RDF-like triples from an object representing a table (within the *Logical Scale*), also showing an example of objects matching the *Match* clause and the respective transformation (within the *Description Scale*). The regular expressions related to the input patterns are represented in the left side, using dashed boxes to define the scope of each wildcard.

It also illustrates the main difference between the two regular expression wildcards for graphs.

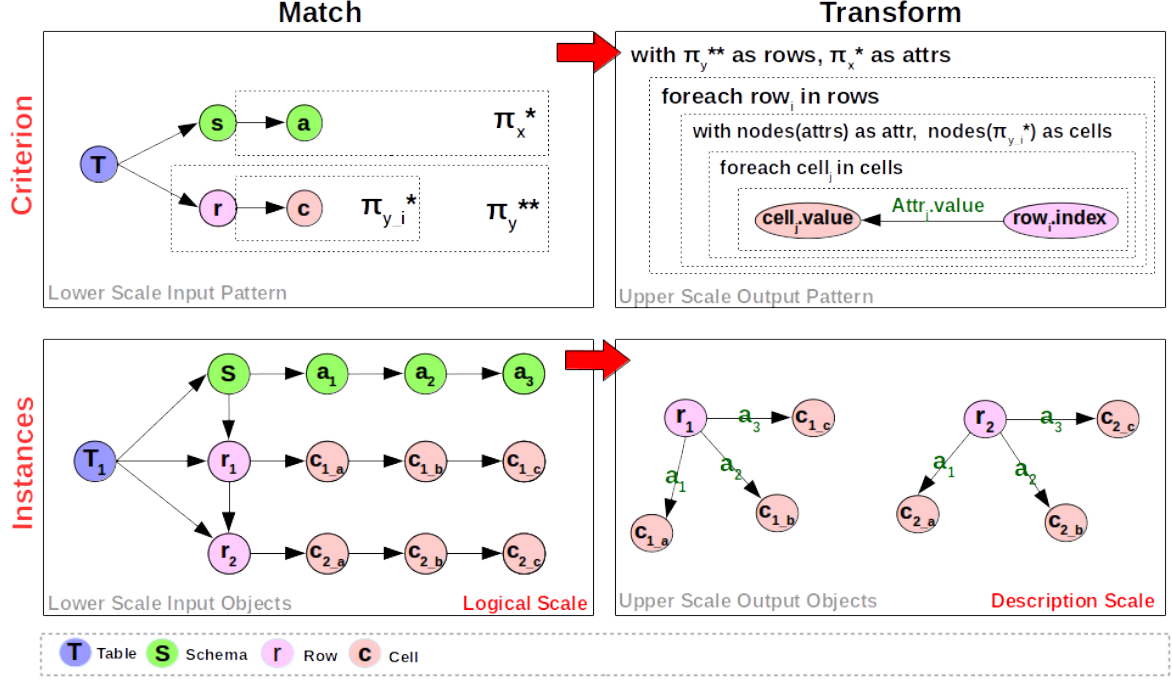


Figure 2.4: Example of a match/transform process

The wildcard  $*$  in  $\pi_x^*$  indicates that each matched object instance can have a sequential repetition of the subpath delimited by the dashed box. The wildcard  $**$  in  $\pi_y^{**}$  indicates that each matched instance can have a parallel repetition of the subpath delimited by the dashed box. The resulting subgraphs are connected to the same origin, i.e., the node  $T$ . The nested pattern  $\pi_{y,i}^*$  indicates a set of connected sequences of nodes  $c$ , where each sequence is connected to the respective origin  $r$  of the outer pattern.

Following the example depicted in Figure 2.4, the *match* pattern is applied to the lower scale containing a graph representation of a table. The pattern  $\pi_x^*$  matches the sequence of attributes of the schema in the row started by node  $s$ . The pattern  $\pi_y^{**}$  matches a set of rows started by nodes  $r$ ; each row corresponds to a line of the table representing a tuple, formed by a sequence of cells matched by the nested pattern  $\pi_{y,i}^*$ .

The right box of Figure 2.4 illustrates the *transform* step of a criterion, using a pseudocode inspired in the Cypher query language<sup>3</sup>. The "with" clause defines a scope, which comprises the set of instances matched by a given pattern. For example, the clause "with  $\pi_y^{**}$  as rows" means that all matched paths for the pattern  $\pi_y^{**}$  will be available in the inner scope of that clause, as instances of a variable *rows*. The inner "foreach" clause navigates through each path  $row_i$  of *rows*. Subsequently, the inner "with" uses the function *nodes()* to return only nodes from the path *attr* and the current  $row_i$ . The innermost "foreach" navigates through all the cells of the row and links

<sup>3</sup><http://neo4j.com/docs/stable/cypher-query-lang.html>

the node corresponding to  $row_i$  with a node representing the value of the cell using the corresponding attribute label.

### 2.4.3 Multiscale Transformations and the Transformation Graph

For each pair of consecutive scales, there is an orthogonal graph linking the objects of the lower scale to the respective derived objects of the upper scale. The objects of the lower scale are subgraphs defined by the match clause of the criterion, as well as objects of the upper scale are the respective derived subgraphs. Such orthogonal graph is disjoint from the graph containing the data in the scales, and is called Multiscale Transformation Graph (MTG). The MTG fosters traceability of transformations along the integration scales, allowing analysis of provenance, reproducibility, reuse, *etc.*

MTG adopts elements of the *PROV Ontology* (PROV-O) [49]. *Entities* are the sources/targets of transformation in PROV-O and they correspond to *objects* in our model (*cf.* Figure 2.5). The transformations between an upper and a lower scales are represented as *Activities*, which correspond to a transformation criterion of our model.

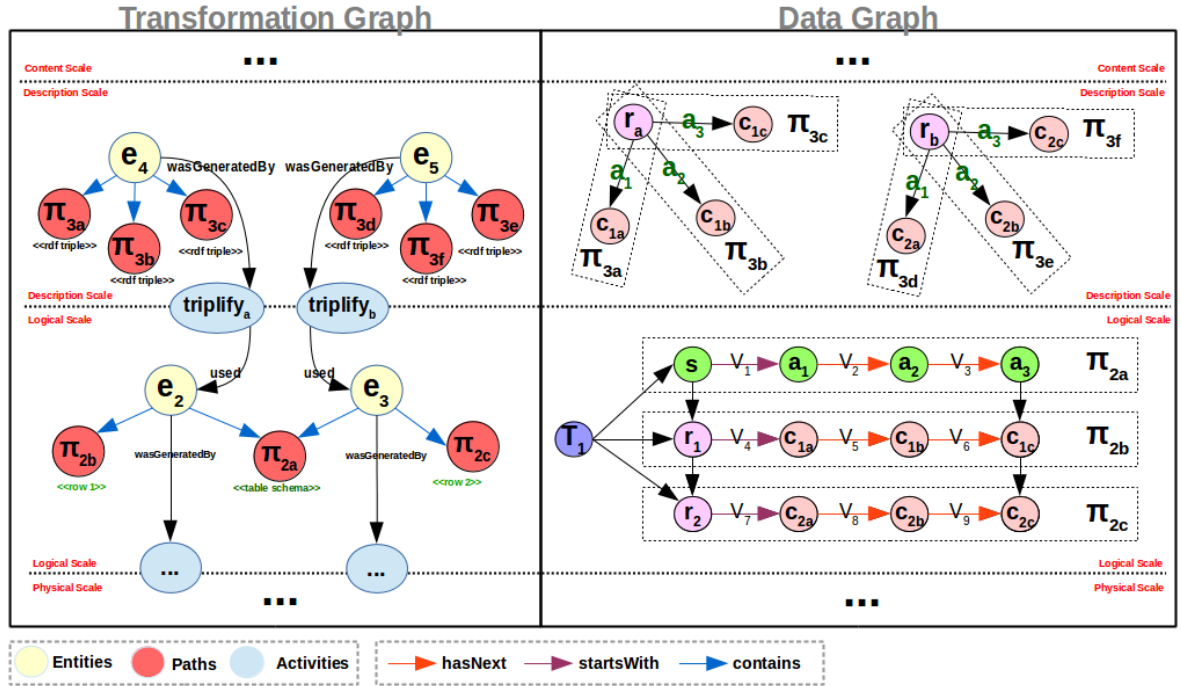


Figure 2.5: Example of transformation between two scales and the corresponding MTG

The example illustrated in Figure 2.5 shows how an instance of a table with a schema and two rows results in two entities ( $e_4$  and  $e_5$ ), each one containing three *paths* representing RDF-like triples. Such transformation is based on a pattern for matching paths in the input and for creating the corresponding nodes and vertices in the output.

The right column in Figure 2.5 (Data Graph) shows an example that meets the patterns specified in Figure 2.4 and its respective MTG in the left column (Transformation Graph). The MTG identifies two *entities* ( $e_2$  and  $e_3$ ) related to the lower scale based on two

respective objects that includes three *paths* ( $\pi_{2a}$ ,  $\pi_{2b}$  and  $\pi_{2c}$ ). The path  $\pi_{2a}$  refers to schema  $s$  of the table. Similarly, paths  $\pi_{2b}$  and  $\pi_{2c}$  refer to the objects representing rows of the table.

The object and respective entity  $e_2$  is composed by the path  $\pi_{2b}$  (the first tuple of the table) and path  $\pi_{2a}$  (the attributes of the schema). The object and respective entity  $e_3$  shares with  $e_2$  the path  $\pi_{2a}$  (attributes of the schema) and also refers to the second tuple of the table ( $\pi_{2c}$ ). The entities  $e_2$  and  $e_3$  are the input for the transformation activities *triplify<sub>a</sub>* and *triplify<sub>b</sub>*, which "triplicates" table rows. The *triplify* activities produce objects represented by entities  $e_4$  and  $e_5$ , which in turn refers to the paths of the output subgraphs.

## 2.5 Experimental Scenario: Organism-Centric Analysis via LinkedScales

In this section, we describe the implementation of the solution and evaluate its application in a biological scenario, exemplifying the transformation between the scales. We present the whole integration process in a practical scenario, going from the sources to the conceptual scale (organism profiles).

### 2.5.1 Implementing the Solution

Several elements and specific technical issues of the proposed framework have being implemented independently [61]. In a nutshell, aspects related to the conceptual level were investigated in [8], while [55] studied how to extract triples as descriptions from different models. Furthermore, [59] examined the problem of handling a multitude of physical formats, converging to a homogeneous one.

Based on the previous implementations, we developed a unified architecture as a framework on top of the Neo4j graph database. A framework called *2graph* for converting resources to graphs in the *Physical Scale* was developed, currently supporting the conversion to graph of CSV, HTML, XML, XLS, XLSX, N3 RDF and ODS – this set of formats was defined as the most relevant formats for biologists in the organism-centric domain. The framework defines a specific module to convert each specialized format to a graph, and was built on top of DDEX [59]. It can be extended by plugging new conversion modules.

The graphs of the Scales and the MTG are stored together within a Neo4j database, but logically separated by a different set of labels on nodes and edges. Similarly, nodes and edges from different scales are stored within the same graph but are logically sliced by properties indicating their scales.

The Neo4j database offers a specific graph query language (called Cypher) that supports both reading (match step) and writing (transform step) clauses. Cypher supports SQL update-like queries, which enables to combine reading and writing clauses to create new graphs resulting from a matched input. We are working to automatically map our generic transformation approach described in the previous section to Cypher queries.

Even though our proposal can be extended to other file formats, we are currently

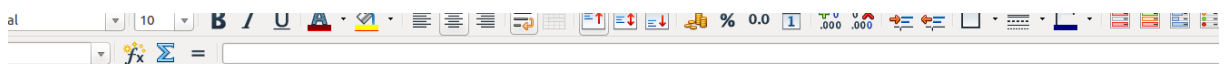
focusing on a set of formats defined by biologists as the most relevant for their work (discussed in Section 2.2.1), *i.e.*, spreadsheets (XLS, XLSX, ODS), HTML tables, CSV files, XML files and textual documents. We have developed a graph framework for ETL named *2graph*<sup>4</sup>. This framework is represented as the “*Graph Translator*” element in Figure 2.2.

## 2.5.2 Scenario and Experimental Procedure

In Section 2.2.1, we presented a scenario of an organism-centric data analysis, in which researchers dynamically produce profiles of living beings integrating characteristics scattered across several sources. The dynamical nature of this task and the heterogeneity of formats, models and schemas on the sources make the progressive incremental integration approach a powerful alternative.

In our investigation, we first collected data corresponding to the biologist’s necessities in the scenario. We applied the implemented tools in these data analyzing the transformation results in each scale. We selected relevant examples to illustrate the findings.

Consider Figure 2.6 and Figure 2.7 with excerpts of files to be integrated: an XLSX spreadsheet and an XML/NEXUS document, respectively. While the spreadsheet contains morphological traits, behavioral aspects, habitat characteristics *etc.* of several species, the XML/NEXUS file corresponds to the serialization of a phylogenetic tree.



A	B	C	D	E	F	G	H	I	J	K	L	M
Species	Class	Family	Terrestrial Biomes	Aquatic Biomes	Wetlands	Other Habitat Features	Sexual Dimorphism	Development - Life Cycle	Mating System	Key Behaviors	Length - average - mm	Mass - average - g
Bombina orientalis	Amphibia	Bombinatoridae	Temperate	Forest	Lakes and Ponds	Marsh	Ectothermic	Female larger	Metamorphosis	Polygynous	950	57,5
Brachycephalus ephippium	Amphibia	Brachycephalidae	Tropical	Forest	Coastal		Ectothermic	Sexes alike	Metamorphosis	Polygynous	950	18
Ceratophrys cornuta	Amphibia	Ceratophryidae	Tropical	Forest	Lakes and Ponds	Marsh	Ectothermic	Female larger	Metamorphosis	Polygynous	110	245
Conraua goliath	Amphibia	Petropedetidae	Tropical	Rainforest	Rivers and		Ectothermic	Sexes alike	Metamorphosis	Polygynous		
Craugastor auquasti	Amphibia	Craugastoridae		Chaparral			Ectothermic		Metamorphosis			
Anaxyrus cognatus	Amphibia	Bufonidae	Temperate	Desert or	Rivers and		Ectothermic		Metamorphosis			
Anaxyrus fowleri	Amphibia	Bufonidae		Savanna or			Ectothermic		Metamorphosis			
Anaxyrus houstonensis	Amphibia	Bufonidae		Savanna or			Ectothermic		Metamorphosis			
Anaxyrus quercicus	Amphibia	Bufonidae	Temperate	Savanna or	Lakes and Ponds	Marsh	Ectothermic	Female larger	Metamorphosis	Polygynous	26	
Anaxyrus terrestris	Amphibia	Bufonidae					Ectothermic		Metamorphosis			

Figure 2.6: Excerpt of a XLS spreadsheet highlighting the row regarding the species *Brachycephalus ephippium*

Both resources contain data regarding the same set of organisms under investigation, being relevant to build organism profiles. While the red box of Figure 2.6 highlights a row of the spreadsheet containing information about the species *Brachycephalus ephippium*, the red box in Figure 2.7 points to an XML element – labeled as *OTU* (Operational Taxonomic Unit) – regarding the same species, representing its node in a phylogenetic tree.

## 2.5.3 Ingestion: From the original sources to the physical scale

The first step involves ingesting raw data from the input resources, converting them to a graph representation – see Figure 2.10(A). The purpose of the *Physical Scale* is to solve a

<sup>4</sup>Available at <http://www.lis.ic.unicamp.br/~matheus/projects/2graph>



```

<meta content="study" datatype="xsd:string" id="meta44232" property="prism:section" xsi:type="nex:LiteralMeta"/>
<otu about="#Tl2514692" id="Tl2514692" label="M4514" xml:base="http://purl.org/phylo/treebase/phylo/taxon/TB2:">
  <meta content="Mapped from TreeBASE schema using org.cipres.treebase.domain.nexus.nexml.NexmlOTUWriter@287322d4 $Rev: 1040"
  <otu about="#Tl252503" id="Tl252503" label="Adelophryne gutturosa">
    <meta content="424083" datatype="xsd:long" id="meta44263" property="tb:identifier.taxon" xsi:type="nex:LiteralMeta"/>
    <meta href="http://purl.uniprot.org/taxonomy/491140" id="meta44261" rel="skos:closeMatch" xsi:type="nex:ResourceMeta"/>
    <meta href="http://www.ubio.org/authority/metadata.php?lsid=urn:lsid:ubio.org:namebank:28051" id="meta44260" rel="skos:c
    <meta href="http://purl.org/phylo/treebase/phylo/taxon/TB2:S10202" id="meta44259" rel="rdfs:isDefinedBy" xsi:type="nex
  </otu>
  <otu about="#Tl252522" id="Tl252522" label="Brachycephalus ephippium">
    <meta content="156198" datatype="xsd:long" id="meta44287" property="tb:identifier.taxon" xsi:type="nex:LiteralMeta"/>
    <meta href="http://purl.uniprot.org/taxonomy/164302" id="meta44285" rel="skos:closeMatch" xsi:type="nex:ResourceMeta"/>
    <meta href="http://www.ubio.org/authority/metadata.php?lsid=urn:lsid:ubio.org:namebank:2475617" id="meta44284" rel="skos:c
    <meta href="http://purl.org/phylo/treebase/phylo/taxon/TB2:S10202" id="meta44283" rel="rdfs:isDefinedBy" xsi:type="nex
  </otu>
</otu about="#Tl244097" id="Tl244097" label="Anaxyrus callidryas">

```

Figure 2.7: Excerpt of a XML/NEXUS file highlighting the species *Brachycephalus ephippium*

common initial issue in the data integration pipeline: homogeneous access. The mapping process preserves in the graph as much original format-related information as possible, without homogenization/standardization concerns. For instance, unlike a text-plain CSV file, proprietary spreadsheet formats have substantial extra information, such as, meta-data, comments, text formatting, formulas, links, *etc.* In the current implementation, the ingested graphs are stored in a graph database and can be reached by a graph query language. The ingestion module in the system is conducted by the *2Graph* software, as described in Section 2.5.1.

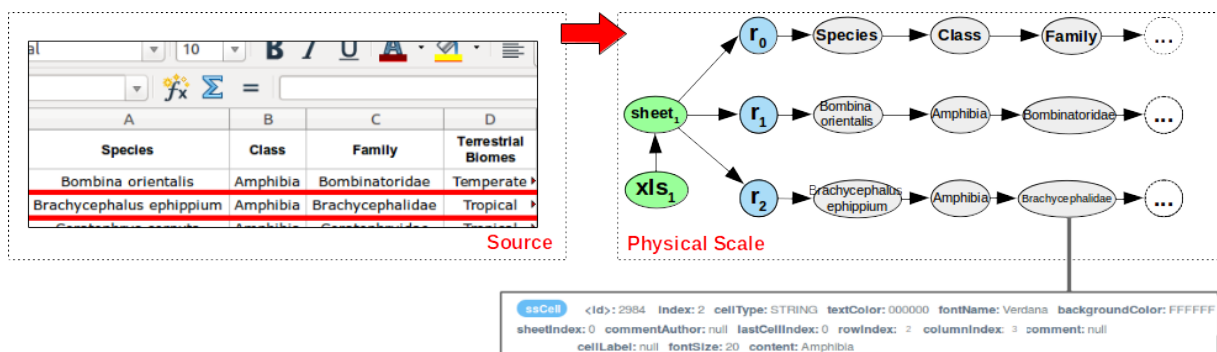


Figure 2.8: Graph Representation of an XLS file as a graph in the Physical Scale

Figure 2.8 and Figure 2.9 depict portions of the mapped graphs produced from the spreadsheet and XML/NEXUS presented in Figure 2.6 and Figure 2.7, respectively. The root node (green) in Figure 2.8 represents a given XLS spreadsheet. It contains a single sheet, which has several rows ( $r_0$  to  $r_2$  in blue). Each row node points to its chain of cell nodes (gray). The box linked to the cell *Brachycephalidae* shows the variety of node properties, representing different aspects of the cell: location, content, format, *etc.* Similarly, the root node in Figure 2.9 represents the XML resource itself, followed by an hierarchy representing the XML document. The highlighted red box represents the XML element *OTU*, previously presented in figure Figure 2.7.

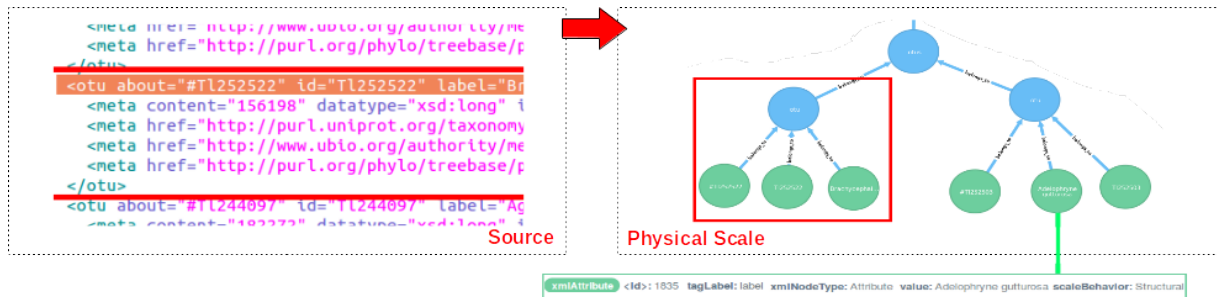


Figure 2.9: Graph Representation of an XML/NEXUS file as a graph in the Physical Scale

### 2.5.4 From the Physical to the Logical scale

Once the resources are represented as graphs in the *Physical Scale*, the integration process starts, and further scales are built on top of it as in a layered architecture. The subsequent *Logical* scale addresses the issue of handling a multitude of formats in a homogeneous logical structure. Transformations between the scales are based on criteria, which comprise a set of match/transform clauses, as detailed in Section 2.4.2. Figure 2.10(A) to (B) illustrates the transformation from the Physical to the Logical scale.

While several formats organize their data as tables and relationships – *e.g.*, XLS, ODS, CSV and even an HTML table –, other organize the data as hierarchies – XML and JSON. Thus, it is possible to induce a common logical representation shared by several physical formats, which aligns or discards unmatched specificities.

Figure 2.10 illustrates the XLS file in the *Physical Scale* and its corresponding representation in the *Logical Scale* as a table structure. While in the *Physical Scale* an XLS format is represented as a grid of cells, with specialized metadata concerning formulas, format *etc.* and no explicit schema – as usual in spreadsheets –, at the *Logical Scale* all *Tables* must look the same, *i.e.*, as illustrates Figure 2.10, the first row of nodes connected to the *Table* node is an explicit Schema defined by its attributes.

The main benefit resulting from the effort of homogenizing multiple formats behind the same logical model is the possibility of reusing algorithms over the same logical structure, independently of its physical format – *e.g.*, the same algorithm can extract entities from tables coming from spreadsheets, CSV, relational tables and others. This transformation rise several challenges – *e.g.*, schema recognition is not always trivial. Such challenges, however, are already widely discussed in the literature (including a previous work developed by us [8]) and are not subject of attention in this research.

### 2.5.5 From the Logical to the Description scale

The *Description Scale* aims at decoupling data from different logical structures and converges them to one single unified logical model. The unified model is based in the triple <resource, property, value>. It relies on RDF, but it still not a full fledged RDF, since it adopts only the RDF graph model to reduce all logical models to a single one. But the content of the nodes and edges are still plain text, lacking fundamental semantic concerns

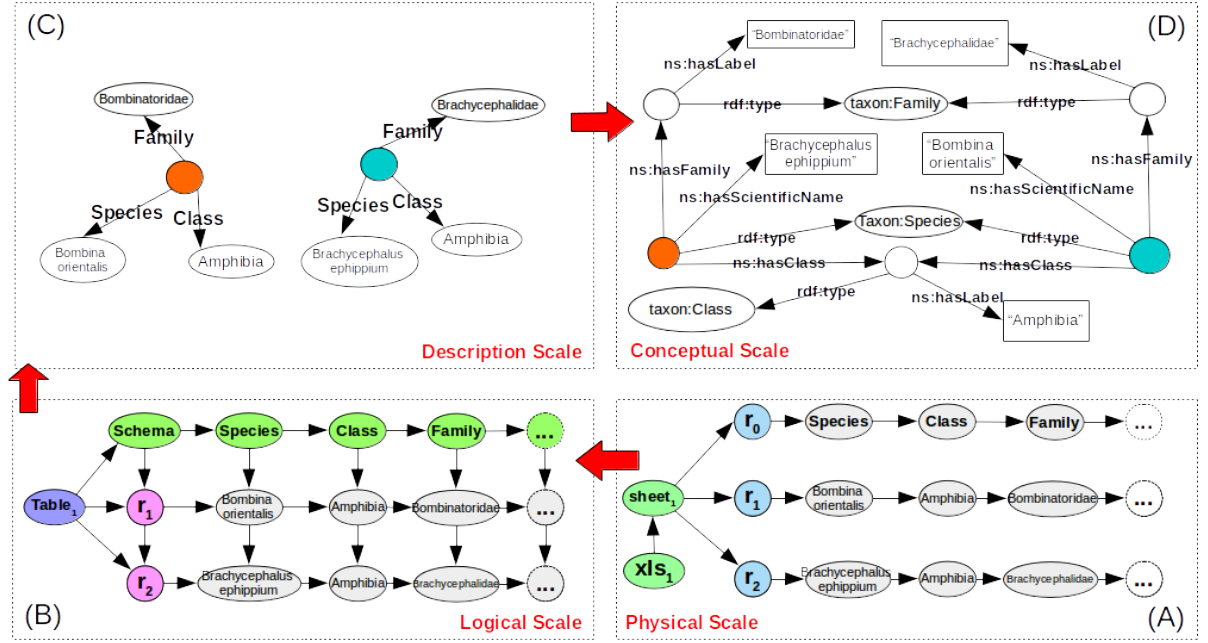


Figure 2.10: All stages presented as a graph-based representation

since it does not: distinguish entities, adopt controlled vocabularies to represent descriptive properties or make explicit the semantics of the elements using ontologies. These issues are addressed in the *Conceptual Scale*.

Initiatives found in literature stress different strategies for transforming a table or a hierarchy to triples, including a previous work developed by us [8]. This research do not focus on such problems and adopts a classical "triplification" strategy – as described in [8]. The transformation approach follows the same rationale of the previous section, to transform data represented as a *Table* in the *Logical Scale* to an RDF-based graph in the *Description Scale*.

The criterion applied in this transformation was described in Section 2.4.2 and illustrated in Figure 2.4 (up), showing the match expression on the left and the transform process on the right. Figure 2.4 (down) shows a materialization of the match/transform: each table row ( $r_1$  and  $r_2$ ) becomes a described instance, in which the descriptive attributes ( $a_1$ ,  $a_2$  and  $a_3$ ) come from the *schema* row and their values ( $c_a$ ,  $c_b$  and  $c_c$ ) come from the rows content. Figure 2.10(B) to (C) illustrates the transformation applied to our frogs example.

Although biologists still cannot handle data from previous scales in a conceptual and more integrated fashion, the *Description Scale* can be helpful to them, as it already allows some preliminary and meaningful analysis. For instance, spreadsheets regarding morphological traits usually adopts a cross-sheet way of organization. Such organization hampers an unified view of the traits of an organism, requiring more efforts from the biologists when conducting any initial analysis.

At this stage of the investigation, *LinkedScales* enables to integrate XML files containing phylogenetic trees (from the *TreeBase* repository) with spreadsheets and CSV files

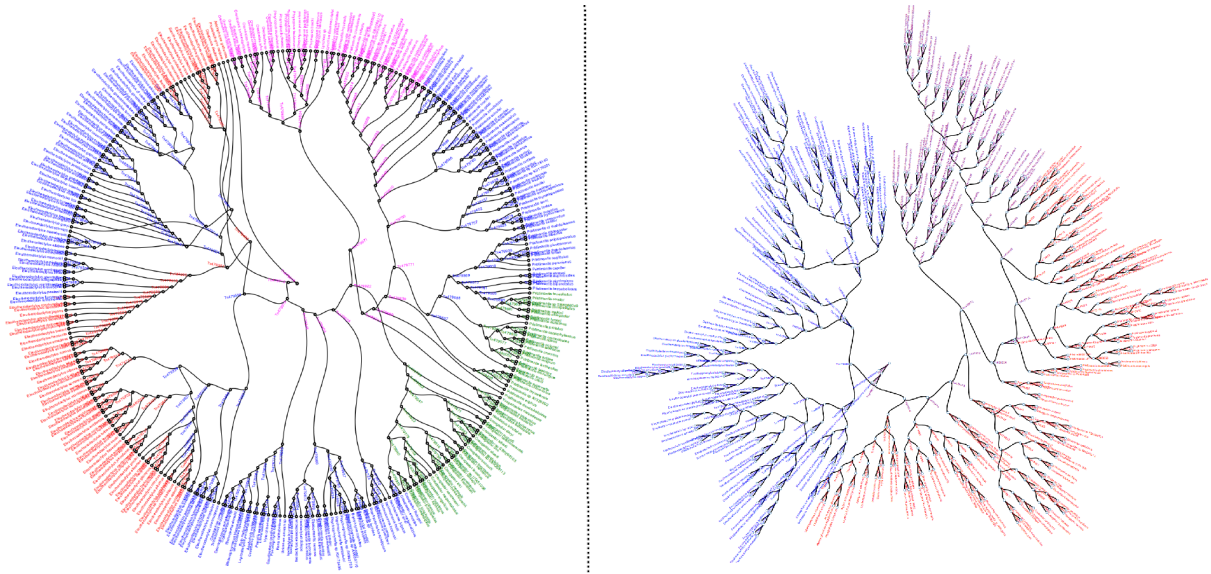


Figure 2.11: Example of visualization of the Description Scale

regarding morphological traits (maintained by biologists). Based on the homogeneous models produced for the files in the *Logical Scales* (after being represented as a raw-format in the *Physical Scale*), species names mentioned on the tree and species names mentioned on the tables are linked using a simple string match.

Figure 2.11 illustrates a visualization of output results corresponding to the initial outcome from the *Description Scale*. It shows the species following the phylogenetic tree provided by the XML file aggregated (colors) according to the tables in which the species are mentioned. Such tree enables the study of the evolution of traits across the phylogenetic group considered, but also correlates how closely related taxa are from one-another.

### 2.5.6 From the Description to the Conceptual scale

The *Conceptual Scale* achieves a full fledged RDF representation. The transformation from the Description to the Conceptual Scale involves applying algorithms like entity resolution and interconnection with ontologies to make explicit the semantics of the entities and properties involved in the description. Therefore, as illustrates Figure 2.10(C) to (D), attributes are unified in the same RDF properties (*e.g.*, `taxon:Species`, `taxon:Family`); entities, like the class *Amphibia* and the family *Brachycephalus*, are unified.

## 2.6 Conclusion

In this article, we proposed an original framework named *LinkedScales*, based on the multiscale integration approach. Its architecture relies on graphs and systematizes in layers (scales) progressive integration steps based in graph transformations. *LinkedScales*

is strongly related with the pay-as-you-go integration, slicing and encapsulating tasks concerned with the integration process in discrete scales. The approach is thus aligned with the modern perspective of treating several heterogeneous data sources as parts of the same dataspace, addressing integration issues in progressive steps, triggered on demand.

The designed solution is based on our Multiscale Graph Model, which was instantiated in our Primary Data Architecture able to be extended to several contexts. The proposal allowed a homogeneous perspective of data in each scale, encapsulating details about heterogeneities. In a nutshell, our approach is founded in three pillars: systematization, reuse and provenance.

The investigated experimental scenario demonstrated the overall potential benefits of *LinkedScales* to reach organism profiles. A significant part of the biological research work remains in an organism-centric perspective, which usually requires combining data regarding distinct aspects of organisms. However, relevant data is typically scattered among heterogeneous sources with different formats, structures and schemas, hampering the combination of data across sources to perceive information meaningfully and to systematically compare organisms. The solution proposed in the *LinkedScales* approach revealed its usefulness to the experimented analysis.

Future work involves conducting additional experimental evaluations to thoroughly examine the quality and scalability of data integration provided by the approach. Furthermore, a full-stack implementation integrating all the independent solutions in an unified system<sup>5</sup> will be developed.

---

<sup>5</sup>For progress, refer to: <http://linkedscales.lis.ic.unicamp.br>

## Chapter 3

# Progressive Data Integration and Semantic Enrichment Based on LinkedScales and Trails

### 3.1 Introduction

Biologists often conduct organism-centric analysis in which organisms – *i.e.*, species or taxonomic groups – are the central focus and data are collected and integrated around them. In this context, biologists might compare organisms in a systematic way and investigate conditions related to their hypotheses. In this context, the construction of *profiles* [81] as "*views*" of data is usual in an organism-centric research. It involves combining data usually fragmented in heterogeneous sources, requiring efforts to collect and combine pieces coming from multiple repositories and files with different formats. The manual process requires a lot of time to prepare data from each source and to integrate them before any analysis. Fig. 3.1 presents a practical scenario where the analysis is based on profiles comprising ecological traits and morphological data. It requires the combination of data from several resources scattered in digital repositories. In this case, the data comes from research repositories associated with scientific publications, such as *Dryad* (<http://datadryad.org>) and *Figshare* (<http://figshare.com>). The combination of datasets is challenging since the different kinds of heterogeneity, *i.e.*, distinct formats (CSV, Excel, NeXML), structures (tables, trees) and schemas, *etc.* require several steps of integration.

Heterogeneity hampers a unified exploration of knowledge across distinct systems. To provide an on-demand lightweight integration, we have defined the *LinkedScales* architecture [56], which aims at splitting the integration steps as discrete scales. Each scale encompasses common aspects and routines related to a particular integration step. *LinkedScales* comply with the dynamicity of modern integration environments, against the classic heavyweight upfront techniques.

This incremental process also produces three kinds of intermediary outcomes: semantic representations, knowledge discovery results and user feedback. They have operational purposes and drive transformation tasks in the production of content in the upper scales. However, there is no a systematic method to record and keep track of these intermediary

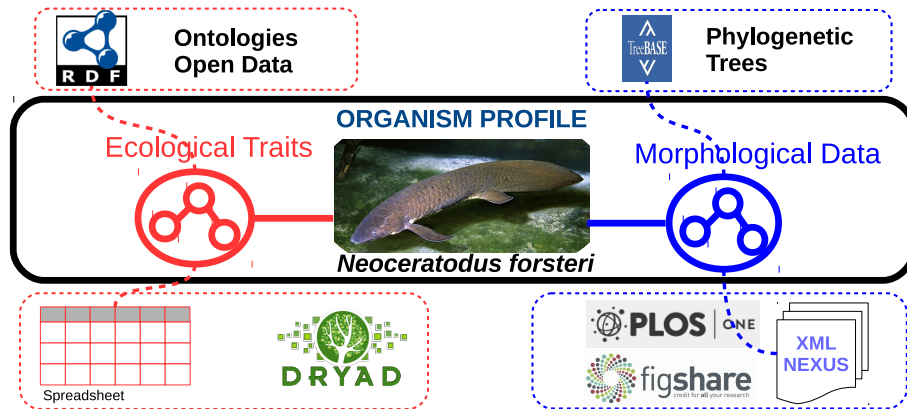


Figure 3.1: Profile integrating characteristics scattered across several sources

outcomes. Operations built over them, like transformation and enrichment, can be better specified, managed and followed if they rely on a standard mechanism to document the outcomes.

In this article, we propose combining *LinkedScales* with the concept of trails. Trails are “*hints*” represented as structured semantic annotations concerning operational scale aspects – *i.e.*, each scale emphasizes a particular step of the integration chain, therefore each scale has distinct types of trails. Trails play the role of metadata associated with portions of data [6]. When trails are included in a progressive integration process, they standardize the way in which intermediary results are represented, which might improve the specification of transformation rules. Furthermore, *LinkedScales* produces a provenance graph while transformations are executed. This graph contains not only information about processes, but also which operational evidence (trails) were considered during the transformation.

We present a practical scenario of exploring trails with *LinkedScales*. We conducted an experimental analysis considering the integration and semantic enrichment of resources related to a particular organism profile. In particular, trails are exploited to guide the process of linking content in the scales with external knowledge bases, like *DBpedia*, to better characterize the data conceptually. In order to show how trails can improve the linking process, in the first step of our experimental procedure, we apply the transformation without the trails and we compare the results taking the trails into account afterward.

The remaining of this article is organized as follows: Section 3.2 presents foundations and related work. Section 3.3 describes the *LinkedScales* framework while Section 3.4 details the proposal of combining *LinkedScales* with trails. Section 3.5 presents the conclusion remarks.

## 3.2 Foundations and Related Work

Several data integration approaches have emerged, including federated databases, schema integration and data warehouses [69]. They mostly rely on providing a virtual unified view under a global schema (GS) [75]. Within GS-based systems, data stay in their original data sources – *i.e.*, maintaining their original schemas – and are dynamically fetched and

mapped to a global schema [35]. It requires a big upfront effort to produce a global schema definition, which may become impracticable due to the inclusion and changes in schemas. Such classical data integration might successfully work when integrating modest numbers of stable databases in controlled environments.

Scenarios in which schemas often change and new data models must be considered still lack an efficient solution. To this end, pay-as-you-go integration approaches implement incremental integration based on progressive steps to continuously refine and improve the connections among sources. The proposal of *dataspaces* aims at providing the benefits of the classical data integration approach but in a progressive fashion way [75]. Dataspaces approach for data integration can be divided into a bootstrapping stage and subsequent refinements. Progressive integration refinements may rely on structural analysis, on user feedback or on manual/automatic mappings among sources [6].

This investigation explores the concept of trails in a pay-as-you-go integration approach. Trails are keyword-based annotations that relate concepts to data sources to be integrated. They are used for a gradual improvement of integration among sources [78]. Trails play a key role since an important step in integration tasks involves defining semantic equivalences across distinct data sources during the dataspace improvement. In some proposals, the user is engaged in helping the semi-supervised process of discovering, suggesting and evaluating mappings, either by statistical techniques or driven by ontologies and dictionaries [6].

As an alternative for the one-step approach to define equivalences between distinct data source elements, trails rely on *services* to support incremental refinements of mappings between schemas. Whenever the user feeds the system with new “hints”, it exploits them to improve the semantic equivalences discovery. These “hints” are treated as a lightweight mechanism to define declarative relationships between loosely integrated data sources [6]. Trails can be associated with either a particular portion of the data or the whole dataset. They can be either automatically inferred or manually assigned, depending on the effort that users are willing to spend [78].

### 3.3 LinkedScales

*LinkedScales* [56] refers to an architecture that systematizes the progressive integration steps, bringing the proposal of multiscale to the data integration chain. It is based on an abstract model that organizes the integration steps as a pile of scales, where the entities in an upper scale are built based on transformations over entities of a lower scale.

The integration starts on the lowest scale, where all original data sources are ingested and transformed into graphs. Each subsequent scale from this point is a graph derived from the previous scale, taking advantage of the flexibility of graphs to logically represent different structures along the scales. This model allows representing operations within and across the scales as transformation procedures in graphs. Fig. 3.2 presents the four scales aiming at going from the raw data sources (lower scales, containing more details about format and structure) to a conceptual scale (fewer details of format and structure, and focus on domain-specific concepts). Scales are interconnected by an orthogonal graph,



supporting traceability among them – *i.e.*, it is possible to "track" sources/targets of transformations between scales.

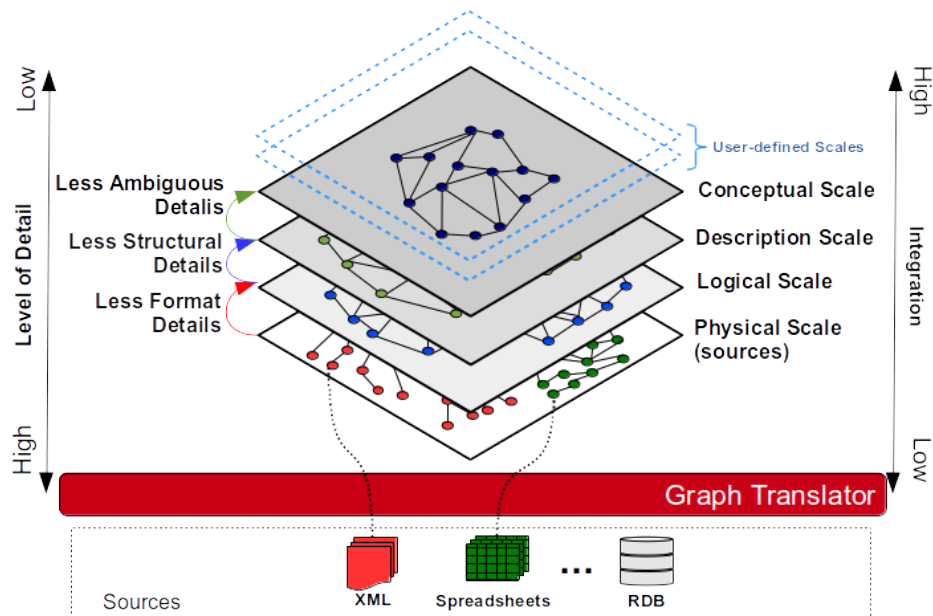


Figure 3.2: *LinkedScales Primary Data Architecture* [56]

The **Physical Scale** aims at representing the different data sources in their original physical format as a graph. The original raw data sources are transformed into a graph by an ingestion procedure. The Graph Translator reads several specialized formats – *e.g.*, Excel, CSV, relational tables, XML – and converts them to an equivalent graph representation. The original structure, format and content of the underlying data sources are reflected in a graph.

The **Logical Scale** offers a common view for data inside similar or equivalent logical models represented in the previous scale. Tables and hierarchical documents are examples of logical models present in the sources. In the previous scale, differences might exist in the representation of a table within a PDF, a table from a spreadsheet and a table within an HTML file since they preserve specificities of their formats.

The **Description Scale** emphasizes the content (*e.g.*, labels of elements within an XML document or values in spreadsheet cells) and their relationships. Since models represent relations among data elements in different ways – *e.g.*, a row in a table can represent data concerning the same entity while hierarchical relations in a document represent aggregations – the *Description Scale* reduces all logical models to a single unified one, to shift the focus towards the descriptive content. The unified model selected for this scale relies on the triple  $\langle \text{resource}, \text{property}, \text{value} \rangle$ , which is usual in several meta-data standards as *Resource Description Framework* (RDF).

The highest scale refers to the **Conceptual Scale**. It integrates data from the lower scale at a semantic level by exploiting the content and relationships between nodes to discover and make explicit the semantics through ontologies. Entities are discovered, deduplicated and related to ontologies as instances of classes, or properties and their values. A “textual graph” of the previous scale becomes a graph containing interrelated

entities and their properties/values, with explicit semantics supported by ontologies.

### 3.4 Combining Trails with LinkedScales

This work involves an enhancement of the *LinkedScales* framework to incorporate *Trails* as the driving component for transformations and provenance. It treats trails as scale-specialized operational semantic annotations, which indicates the role of data portions. Such *hints* are considered by scale transformation processes, incrementally conducting the refinement of the dataspace.

We conducted an experiment in the organism-centric scenario to investigate how trails improve transformations between scales. We collected two complementary sources coming from different scientific publications – as illustrated in Figure 3.3. The first source is an XLS spreadsheet [52] shared in the Dryad repository. The second source is a NeXML file – an XML-based format for representing phylogenetic and phenotypic data, shared in the Figshare repository [66].

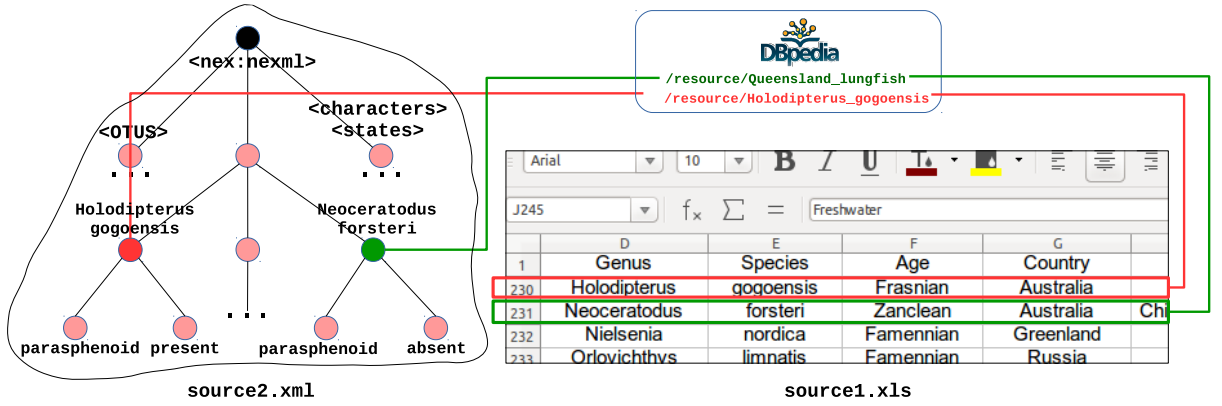


Figure 3.3: Schematic illustration of the bootstrap phase of the experiment

Both data sources are concerned with information about lungfish. While the first data source contains morphological traits, behavioral aspects, habitat characteristics, *etc.* of several lungfish species, the second data source comprises a phylogenetic tree and a phenotypic description in a character/character state format. Even though both data sources are available for researchers, integrating such information conceptually by combining data of the same lungfish species remains a challenging laborious task.

In next sections we exploit the data sources as a running example to describe the use of different types of trails and their relationship with scales. Trails vary according to each scale, indicating relevant aspects of data that the transformation process takes into account during the production of an upper scale. We further describe roles of trails presenting the scale that they are inserted accompanied by the target transformation scale –*e.g.*, a *physical-logical trail* refers to a trail to be inserted in the physical scale, impacting in the data production of the logical scale.

### 3.4.1 Physical-logical Trails

Lowest part of Figure 3.4 presents an excerpt of an XLS spreadsheet containing information from a study of discrete characters change in the evolution of lungfish (class *Sarcopterygii*) [52]. The dataset is an asset associated with a publication, shared in the *Dryad* repository. It describes information about taxonomic classification, associated geological age, type of habitat, countries, *etc.*

Data is ingested into *LinkedScales* database as a graph. The middle part of Figure 3.4 shows partial representation of the ingestion result in the *Physical Scale*. Rows of nodes represent rows of the spreadsheet and their stream of cells. The graph focus on representing as much information as possible of the raw resource. Via such data, the logical organization can be inferred or derived – *e.g.*, initial and boldly formatted cells usually are the table schema.

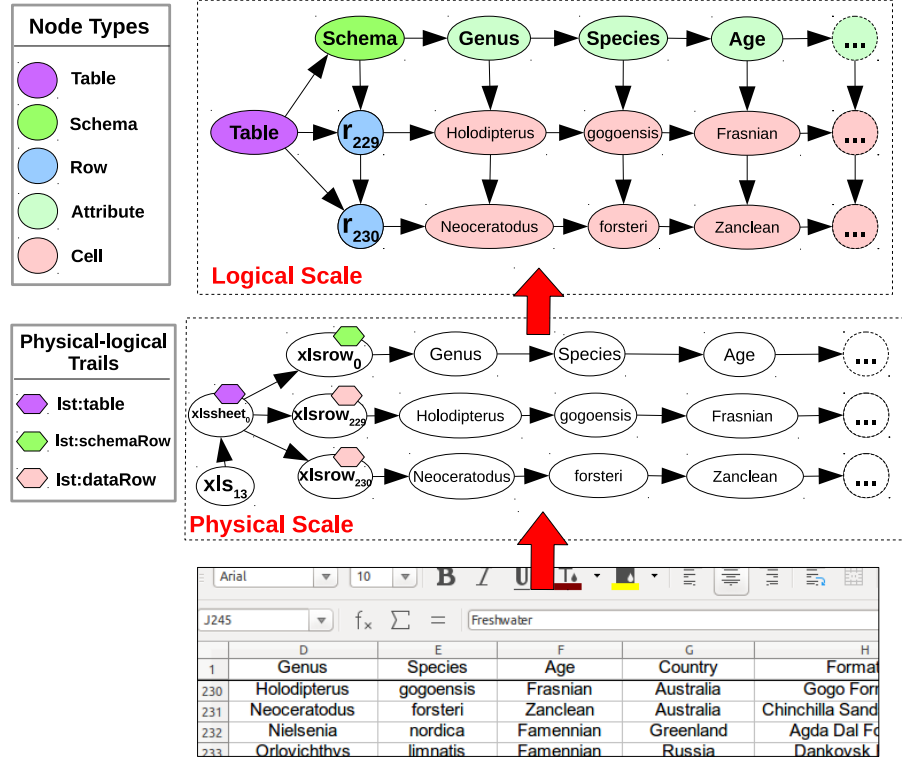


Figure 3.4: Excerpt of a XLS and its representation on the *Physical* and *Logical scales*

**Physical-logical trails** – pictured as colored hexagons in Figure 3.4 – are inserted to distinguish types of structures and their internal components. Figure 3.4 (middle part) illustrates how trails are used to conduct transformations from the physical to the logical scale. Trails associate structure-related roles to the nodes as: table (*lst:table*), row (*lst:dataRow*) and the stream of cells corresponding to the schema (*lst:schemaRow*). In the bootstrap phase of the dataspace, this type of trail is either automatically inferred by the ingestion module, according to the internal structures, or specified by the user. In short, the Physical-logical trails indicate how data is logically organized within the format-specific graph representation of the resource.

Based on the associated physical-logical trails, a transformation process adopts a standard representation of structural elements of *tables* to logically represent the resource in

the logical scale. Representing structures using a standard representation in the logical scale is particularly important, as it allows, for instance, reusing table-related algorithms to reach resources independently of formats.

### 3.4.2 Logical-description Trails

The *Description Scale* aims at shifting the focus to the content and their relationships, reducing logical models to an RDF-based structure. The bottom part of Figure 3.5 illustrates how **logical-description trails** are used to produce the description scale from the logical scale. At this point, trails indicate how structural elements should be organized as  $\langle \text{resource}, \text{property}, \text{value} \rangle$  triples.

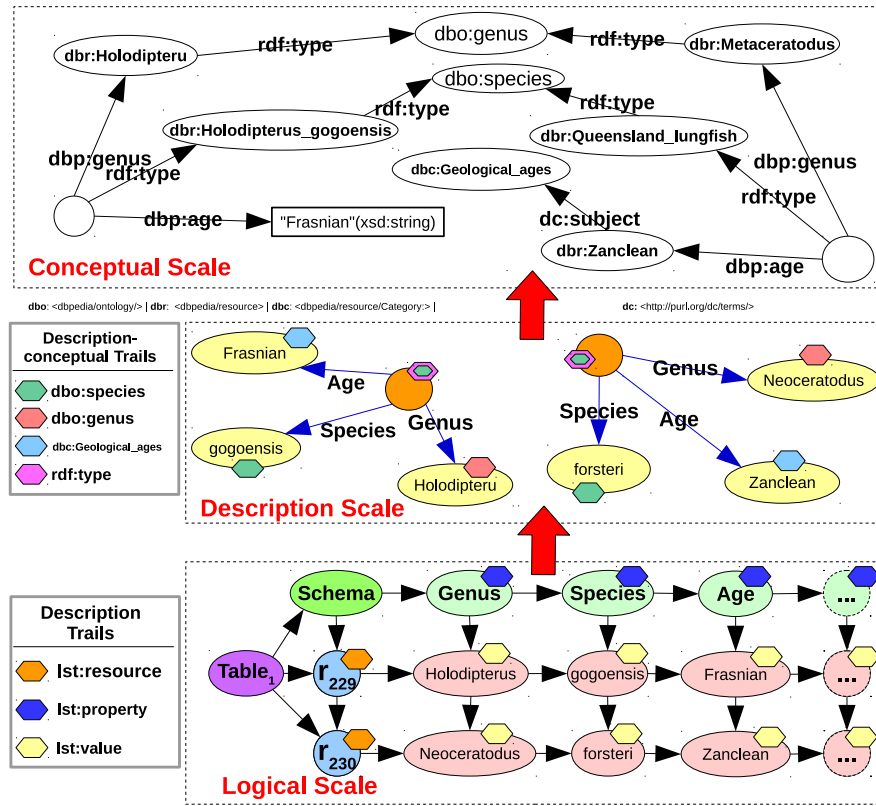


Figure 3.5: Logical-description trails driving a logical-description scale transformation, and description-conceptual trails driving the production of the conceptual scale

Figure 3.5 illustrates how trails (colored hexagons) are associated with structural elements on the *Logical Scale*, indicating, for instance, that rows (nodes  $r_{229}$  and  $r_{230}$ ) are resources, schema attributes (green nodes *Genus*, *Species*, *Age*) are properties and cells are values – e.g.,  $\langle r_{230}, \text{Genus}, \text{Neoceratodus} \rangle$  and  $\langle r_{230}, \text{Species}, \text{forsteri} \rangle$  are triples produced based on trails.

The transformation illustrated in Figure 3.5 can be represented by a rule which matches a pattern (including specific trails) as input and produces a transformed output. Transformation patterns are already defined in the *LinkedScales* model [56], and are beyond the scope of this work.

### 3.4.3 Description-conceptual Trails

**Description-conceptual** trails focus on reaching an expected perspective – e.g., organism profiles. Figure 3.5 (upper part) illustrates trails indicating the expected semantic interpretation of nodes in the description scale, making the semantic explicit by adopting specific elements of ontologies. Such trails can be automatically discovered by the system in a semi-supervised process or be directly assigned.

The *Conceptual Scale* addresses fundamental semantic concerns by distinguishing entities and adopting controlled vocabularies to represent descriptive properties. Adjustments – removing or adding description-conceptual trails – made on previous scale are a way for handling the dynamicity of scenarios as organisms-centric research in terms of testing different hypothesis.

Scales and trails play complementary roles in the progressive integration process. While a scale provides a homogeneous view of the lower layers, trails offer the proper clues for the transformation to the next scale. Consider, for example, the logical scale. It offers a homogeneous view of data considering the logical model, i.e., all tables are represented in the same way, as well as, all trees. If on one hand, this is a powerful mechanism, as the heterogeneity of several table formats is hidden in a lower scale, enabling to reuse the same algorithms for several homogeneous tables, on the other hand, these algorithms need clues to interpret implicit differences which will impact in the next scale.

Regarding the experiment of integrating both XLS and NeXML sources, at the bootstrap stage, after ingesting both data files and converting them to the Logical and Description scales in the graph, we used *DBpedia* (dbpedia.org) to automatically produce the trails that guided the production of the Conceptual scale. The experiment aimed at connecting portions of the data source with DBpedia resources (English release of October 2015), and therefore indirectly linking and enriching similar resources.

Our procedure searches in the *DBpedia* for the most similar resources of each node in the graph. The search method compares the input query against the *DBpedia* resource contents. This comparison uses the *tf-idf* measure and may return approximate/incorrect results like uncorrelated resources. To examine the benefits brought by the trails in this transformation process, the next integration stage inserted trails associated to the nodes to give clues to our integration system about the nature of the nodes in the graph. In this experiment, two trails were considered associated with specific procedures:

**-Species related Trail:** The user tags the nodes that represent *Species*, then the system can filter, via SPARQL queries, the resources returned by the bootstrap stage that are instances of taxonomy-related classes, according to the *DBpedia* ontology.

**-Morphological related Trail:** The user tags the nodes that represent morphological characters. Such trails are used as input in an entity-quality recognition algorithm [65] that extracts morphological characters inside a free-text and creates an Entity-Quality (EQ) representation. The Entity element refers to the morphological character (e.g., *bone*) and the Quality stands for a qualifier (e.g., *present*) that specifies a given state of the Entity. The algorithm uses two domain-ontologies to support its recognition task: (1) Teleost Anatomy Ontology (TAO) to recognize the Entities and (2) Phenotypic Trait Ontology (PATO) to recognize the Qualities.

Figure 3.6 depicts a portion of the conceptual scale with (right part) and without (left part) trails. Each node in the figure represents a specific species from both data sources – the first data source in green, and the second data source in red – and edges represent relationships concerning taxonomy and morphological traits (entity-quality pairs). When trails are associated to elements of the previous scale (description), the produced conceptual scale is semantically refined according to the expected requirements in the organism profiles.

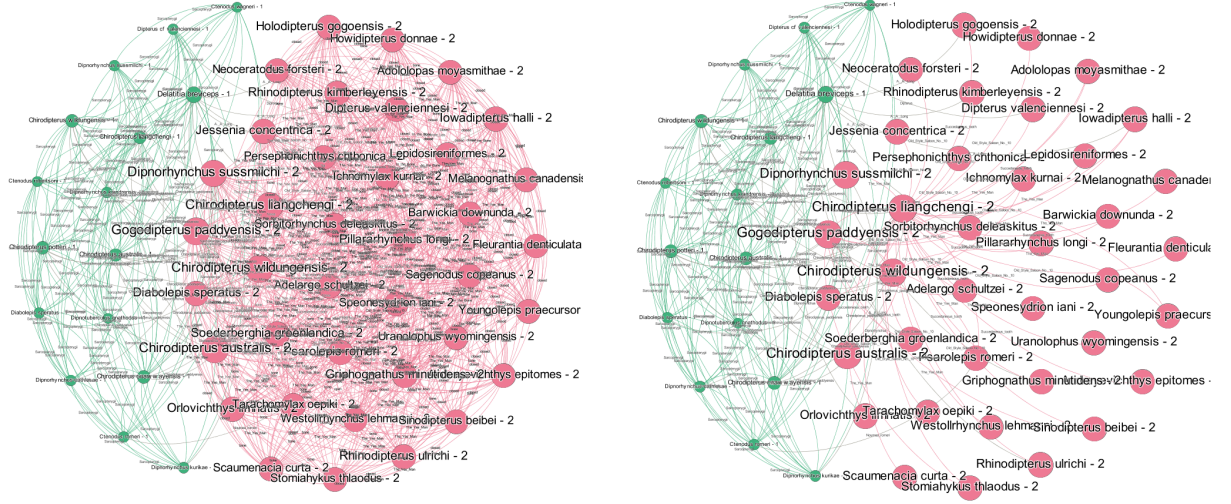


Figure 3.6: Comparison of the same portion of two conceptual scales: Non trail-based (left) and trail-based (right) transformations

## 3.5 Conclusion

A significant part of the biological research work remains in an organism-centric perspective, which usually requires combining data regarding distinct aspects of organisms. In this article, we presented how our *LinkedScales* framework, based on the multiscale integration approach, can work aligned with trails as operational semantic annotations. Trails systematize intermediary outcomes, improving the transformation process and provenance records among the scales. Our experimental analysis demonstrated the overall potential benefits of trails in *LinkedScales* to reach organism profiles. Future work involves conducting additional experimental evaluations to thoroughly examine the quality and scalability of data integration provided by the approach.

## Chapter 4

# Multiscaling a Finding-Disease Dataspace

### 4.1 Introduction

Evidence-based medicine (EBM) is a movement aiming at increasing the use of conscientious and rational clinical decision making, emphasizing the use of evidence from previous, reliable and well-conducted research [73, 70]. Nowadays, EBM is the best approach to developing a therapeutic plan to a patient, since it comprises the best evidence, patient values, and personal characteristics together with clinical experience.

Several articles describing systematic reviews, meta-analyses, and randomized controlled trials are available in the literature in multiple repositories. However, the amount of knowledge available to physicians is increasing exponentially, and external memory devices, such as electronic libraries and artificial intelligence apps are becoming more necessary to assist physicians in their daily activities. Noteworthy, EBM methods are also important to address the clinical reasoning process of reaching an accurate diagnosis for a particular patient [21].

There are two different ways of establishing a diagnosis considering the neurobiological reasoning process: pattern recognition and analytical reasoning. Both thinking processes are based on the complex interconnection of signs and symptoms presented by the patient that are aggregated in the physician's mind through a complex hierarchical chain of interconnections. Traditionally, doctors used to learn to value these signs and symptoms without a scientific approach based on their real epidemiology, but relying on a repository of "collective memories." Recently, inspired by the EBM movement, physicians are concerned about creating an Evidence-Based Clinical reasoning, in which precise epidemiological information is collected to each one of the main complaints, such as sensitivity, specificity, and likelihood ratios, to function as a reasoning map to enhance the accuracy of the clinical hypothesis.

In this context, it is important to prepare the next generation of physicians to use all these strategies which prove to nurture the accuracy of clinical reasoning. Prevention of cognitive errors is crucial to promote patient safety. Therefore, we must struggle to develop teaching strategies for undergraduate and postgraduate medical trainees which

incorporate evidence-based clinical diagnosis and reasoning.

Even though there are efforts to put together and compile data concerning evidence-based medicine as systematic reviews and meta-analyses, there are two main limitations to using them for medical training: (i) the knowledge is segmented according to the illness, but when trying to reach a diagnosis physicians also work with the complaints' scripts which can drive to different diseases; (ii) the evidence-based clinical diagnosis learning process starts by the memorization of several compilations, to be further evoked in real scenarios, when the physician learns how to apply them.

This work is part of a bigger project aiming at training physicians through the exposition to variable simulation scenarios, grouped online in a game format, in which they must diagnose patients in an emergency room. As will be further detailed, the game has been built on top of a knowledge base that combines real cases data and evidence-based data.

This paper focuses on a particular challenge of building the game knowledge base, gathering together data scattered in several heterogeneous sources of EBM, ranging from tables inside PDF files to spreadsheets. We address the problem through our multiscale dataspace architecture, which systematizes each aspect related to extraction, integration, and transformation as layers (scales).

Inspired by the encapsulation principle of multilayered software architectures, each scale hides from its upper scale the specificities of the data it receives as input, presenting a standard interface according to its role. It enables to factor the different aspects of the problem per scale and to reuse algorithms developed for a scale, even when there are changes in lower scales.

The remainder of the paper is organized as follows: Section 4.2 introduces foundations of EBM and shows the relationship with our problem; Section 4.3 presents the related work; Section 4.4 details our architecture; Section 4.5 shows how our architecture is applied in the EBM context; and Section 4.6 presents the conclusions and future work.

## 4.2 EBM Data as a Basis for Medical Training

### 4.2.1 EBM Knowledge Base

Our approach to exploit evidence-based medicine data in medical training involves a simulation game, supported by an evidence-based knowledge base. This base feeds three fundamental operations of the game: guidance/feedback, evaluation, and synthesis. This paper focuses in the process of building such a knowledge base departing from existing evidence-based medicine data, which is scattered among several resources, as will be further detailed. Nevertheless, we start by describing the game and its interaction with the produced base.

Consider a diagnosis of Acute Myocardial Infarction in a patient. One of the key artifacts to support the evidence-based rationale is presented in Figure 4.1. The table is built from a scenario of a particular population of patients presenting specific characteristics – e.g., age range, chief complaint, a context of admission – in which there is a probability of having a diagnosis of Acute Myocardial Infarction (25% in the study of Figure 4.1). The



probability related to this base scenario is known as the pre-test probability [54].

Clinical Feature	Likelihood Ratio (95% Confidence Interval)
Pain in chest or left arm	2.7*
Chest pain radiation	
Right shoulder	2.9 (1.4-6.0)
Left arm	2.3 (1.7-3.1)
Both left and right arm	7.1 (3.6-14.2)
Chest pain most important symptom	2.0*
History of myocardial infarction	1.5-3.0†
Nausea or vomiting	1.9 (1.7-2.3)
Diaphoresis	2.0 (1.9-2.2)
Third heart sound on auscultation	3.2 (1.6-6.5)
Hypotension (systolic blood pressure $\leq$ 80 mm Hg)	3.1 (1.8-5.2)
Pulmonary crackles on auscultation	2.1 (1.4-3.1)

Figure 4.1: For patients presenting with acute chest pain, the clinical features and the respective Likelihood Ratio concerning the probability of a Acute Myocardial Infarction (source [64]).

On top of the pre-test probability, the table presents relevant clinical features and the respective likelihood ratios (LRs). The importance of knowing the LRs of each one of the complaints of the patient is related to its power of modifying the pre-test probability of a particular diagnosis. The probability of the disease enhances or diminishes accordingly with the LR value. If the physician is aware of the pre-test probability, he or she can use the LR in a nomogram to calculate the post-test probability, as shown in Figure 4.2.

A nomogram is this diagram representing the relation among three variables through scales: pre-test probability, likelihood ratio and post-test probability in our case. In the example of Figure 4.2, for a pre-test probability of 25%, an LR of 7.1 (clinical feature "chest pain radiation Both left and right arm") results in a post-test probability of 70%. Although most of the time doctors do not use the nomogram, they can at least create a hierarchy of probability powers to help analyze the clinical problem in focus, which can improve diagnostic accuracy.

Tables like this are available for specific diseases and their possible complaints. Our game starts by presenting a chief complaint and extra information to the player, who will try to figure out the possible diagnosis, to proceed the next steps. For example, chest pain can indicate Acute Myocardial Infarction (likelihood rate of 2.7), but can also indicate Pulmonary Embolism, Tension Pneumothorax, Aortic Dissection, or Esophageal Rupture. This kind of information involves a compilation of information available in several tables like Figure 4.1, as presented in the graph of Figure 4.3.

The graph is a fragment of our knowledge base that starts with a chief complaint – chest pain (on the left). It conducts to scenarios and the respective pre-test probabilities. Related to each scenario, there are relevant clinical features and their associated LRs when considering a given diagnosis.

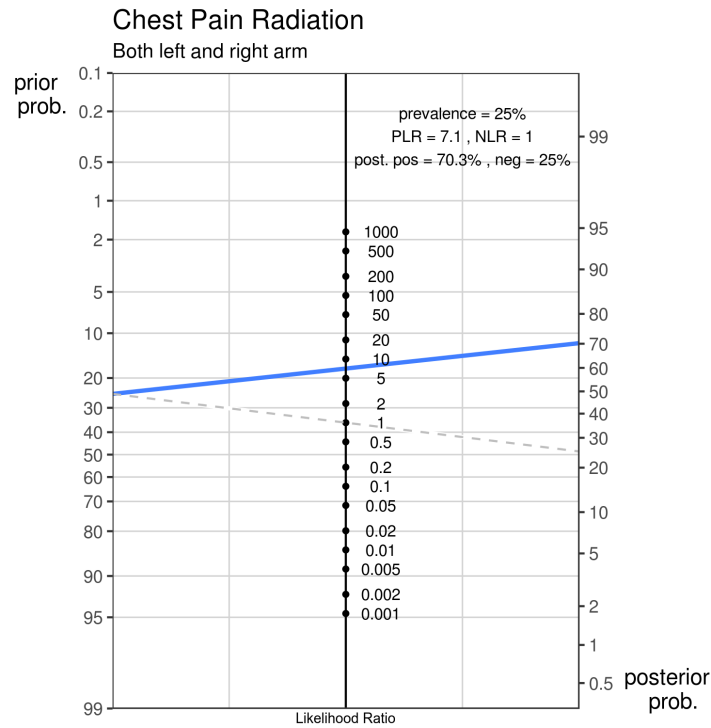


Figure 4.2: Nomogram applying the LR of 7.1 over the probability of 25%.

#### 4.2.2 Game and Knowledge Base Interaction

The game developed here is based on a previous successful experience of an Emergency Medicine course developed and conducted in the Faculty of Medical Sciences at University of Campinas (Unicamp) [18]. It starts by presenting an initial profile of a patient admitted to the emergency unit of the hospital. The case evolves in virtual rounds, in which students receive updated bulletins of the case, with textual descriptions and images, and must address questions and challenges presented by the course teachers. All interaction is through a forum of the Moodle platform.

The current project is transforming the course in a simulation game, automatizing as far as possible tasks of guidance, assessment, and feedback. We envisage three advantages in this automation: these interaction tasks of guidance, assessment, and feedback can be performed in real time during the execution of the simulation; it will be possible to scale up the game to lots of players; the interaction is based on an evidence knowledge base compiling previous, reliable and well-conducted research. Moreover, teachers still have a fundamental role in the course, but they can direct their attention to higher level interaction issues.

As in the emergency course, the cases in the game are derived from real patients from the hospital, but all the data is anonymized before utilization. The game can be played in two modalities. Beginners play a guided simulation. This modality proceeds in virtual rounds, as in the emergency course. Departing from an initial profile of the case, emphasizing the initial complaint, the game presents progressive steps of the case, asking a question to the students and requesting actions from them. In each step, the system evaluates the student answer/action and gives him feedback and guidance, based in the

knowledge base, as further detailed.

Consider that the game will simulate a case of a patient presenting chest pain as initial complaint. Patients presenting acute chest pain (ACP) as the chief complaint are a common scenario in emergency departments. Only in the United States, per year, ACP corresponds to 8 million visits to emergency departments [68], representing approximately 10% of all cases [9]. Therefore, training physicians to understand and consider the influence of the evidences they find during their clinical reasoning is fundamental.

Causes of ACP are notably vast, possibly associated with cardiac, pulmonary, gastrointestinal, psychiatric, musculoskeletal and many other problems. Five causes, however, are extremely life-threatening [9, 68] and must be rapidly addressed in an emergency room: (c1) Tension Pneumothorax; (c2) Pulmonary Embolism; (c3) Esophageal Rupture; (c4) Acute Myocardial Infarction; and (c5) Aortic Dissection. During a clinical reasoning, physicians try to rapidly confirm or exclude the causes that pose an immediate threat to life.

A player in the beginner modality can follow the case structure discussed in [18], in which the system asks which can be the life-threatening causes of chest pain. The graph in the knowledge base, whose fragment is illustrated in Figure 4.3, can be exploited to assess how close is the player from the real possibilities and to give detailed feedback.

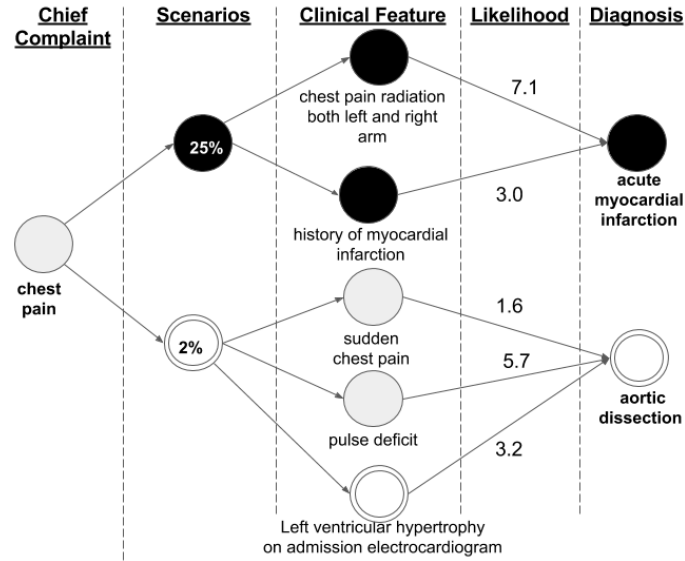


Figure 4.3: Relationships among clinical features and diagnosis compiled in a graph-like structure plus the case planned for the game (black nodes) and the route pursued by the player (white nodes with two borders).

A player in an experience modality will proceed to solve the case as in a real scenario, i.e., collecting evidence and using them to achieve a diagnosis. In a first examination, the patient also complains that the chest pain is radiating to both left and right arm. Considering an initial pre-test probability of 25%, the radiation complaint with LR of 7.1 raises the post-test probability to 70%. The high relevance of this post-test probability justifies following the path of considering Acute Myocardial Infarction, indicated by black nodes in the graph.

The player, however, did not consider Acute Myocardial Infarction as the first hypothesis immediately, but correctly asked for an electrocardiogram (indicated by white nodes with double borders in the graph), which showed left ventricular hypertrophy. As the electrocardiogram did not show a specific finding directly connect to myocardial ischemia, the player wrongly considered Aortic Dissection as the principal hypothesis.

The comparison of the two graphs (the expected and the pursued) will be used to assess the student performance. Moreover, at any stage, the system can give guidance to the student informing, for example, that the pain radiating to both left and right arm has no proper relation with aortic dissection.

This graph illustrating the first step of a patient examination can be extended to further steps. If the clinical features are independent, it is possible to transform a given post-test probability in pre-test to apply the next feature LR. In any case, it is always possible to compare the expected and pursued graphs, to assess and assist the player.

## 4.3 Related Work

### 4.3.1 Computer-based Medical Learning

Health systems for decision support or training have been exploiting knowledge bases for a long time. MYCIN, which was one of the first decision-support systems, express the knowledge as a set of rules [13]. It uses about 500 rules on an *if-then* structure to help doctors identifying and finding the disease-causing microorganisms and the proper antimicrobial therapy respectively. Additionally, it provides an explanation mechanism for the reason and the manner how the system took the decision.

In this work, we are interested in a particular kind of systems, which represent the knowledge as a network of clinical features and illnesses. As we will further describe, these networks are a proper representation to convert existing data in a unified base.

CASNET is a system that infers some prognostic conclusions and diagnoses, considering tests with ophthalmologists in the glaucoma domain [48].

The system works with a casual network illustrated in Figure 4.4. It has four planes. The plane of observations comprises signs, symptoms and test results observed in the patient before interpretation, e.g., a test of visual acuity. The plane of pathophysiological states involves the understanding of abnormal conditions or mechanisms, departing from the observations. They form a causal network in which an edge between two states A and B means that A can cause B. The plane of diseases indicates diseases that are deduced from an observed pattern (causal pathways) in the pathophysiological plane. A fourth layer, not shown in the figure, identifies treatment plans, related with diseases.

Probability-based computational systems which relate features and illnesses for learning have an important agent since the eighties: Internist-1. It has arisen as a decision-support tool in general internal medicine. The system has undergone constant evolution for about 30 years, until the first decade of the years 2000, becoming the QMR system (Quick Medical Reference) [76]. According to [76], the current knowledge base contains 5,000 findings and 700 diagnoses, accumulating 53,000 relationships between them.

By suggesting likely disease candidates, Internist-1 conducts the physician throughout

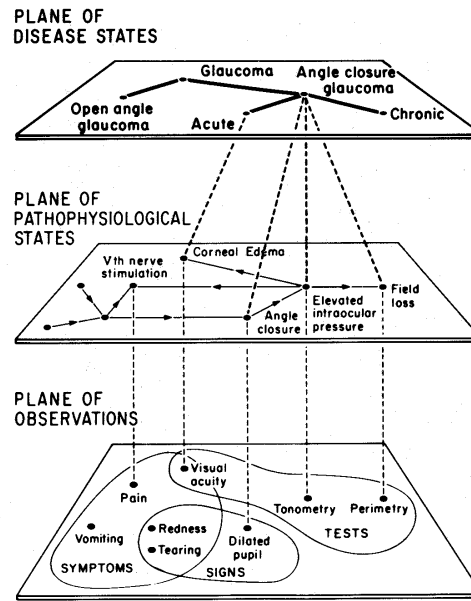


Figure 4.4: CASNET 3D Description of Glaucoma (source [48])

the patient evaluation. It uses a heuristic reasoning method in conjunction with a *quasi*-probabilistic scoring scheme [74]. A *quasi*-probability distribution is very similar to the probability one, but with some relaxations. That is, it considers satisfactory approximate solutions which function for easier problems when handling with more complicated ones.

The Internist-1 system calculates its probabilities by taking into account two properties, *evoking strength* and *frequency*. *Evoking strength* denotes how strongly a physician should take into consideration a diagnosis in the presence of a finding, against all the other diagnostic possibilities. *Frequency* expresses the incidence of a finding in a determined disease.

The Internist-1 reasoning system works with the combination of differential diagnoses to progressively refine and reinforce the conclusion. Given manifestations, the system intersects the related differential diagnoses sets. The *evoking strength* and *frequency* values are recalculated through the Internist-1 reasoning system considering the intersection.

### 4.3.2 EBM and Consensual Knowledge

As far as we know, the existing systems produce their bases from the knowledge of an expert or of a restrict group of specialists (e.g., from a university or research group). No system exploits the collaborative knowledge widely produced by the doctor's community, as the consensual studies systematized in evidence-based medicine. In the available literature, EBM data is usually shared as tables. Such tables, however, are scattered among spreadsheets, HTML pages, and textual documents. Therefore, the reuse and consumption of the conclusions of researches and meta-analyses (a compilation of several third party analyses) is hampered by multiple serialization formats and nonstandardized structural organization.

Figure 4.1 is a table from a meta-analysis PDF file associating clinical features (CFs)

and likelihood ratios (LRs) from multiple pieces of research regarding a specific clinical condition – extracted from [84]. Related data, however, can be found as HTML pages within the EBM-focused repository of JAMA – a peer-reviewed Journal of the American Medical Association (<http://jamaevidence.mhmedical.com>).

### 4.3.3 Classical vs. Progressive, On-demand Data Integration

The increasing need for handling multiple, distributed and heterogeneous data sources resulted in several data integration strategies, such as federated databases, schema mappings and data warehouses [28, 69, 75, 35, 32]. Commonly adopted data integration approaches regularly focus on providing a unified "*view*" of the sources, following a predefined global schema (GS) [75, 46].

GS-based systems preserve original sources, dynamically fetching and mapping requests accordingly to the predefined mapping and the metadata regarding capabilities of the participating resources. Queries sent to the mediator are processed according to the GS, and subsequent queries are then dispatched to the resources. Underlying resources (e.g., a database management system) are abstracted via specialized wrappers [50, 35].

Although the idea of encapsulating multiple sources as a single database is useful to applications, producing the global schema can be a major drawback, as it requires a significant upfront effort [50, 30]. Furthermore, adding a new resource or adjusting the schema of a participant resource may hamper the execution and guidance of the whole mediation process.

Classical data integration approaches might favorably work when integrating databases in domains with already established schemes. Since scheme-static domains are rare, *pay-as-you-go* integration strategies have emerged. Instead of requiring costly, labor intensive upfront mappings between schemas before having access to the data, on-demand integration strategies aim at providing early access to the data via small cycles of integration – *i.e.*, if the user needs the data now, some integration is better than nothing.

The notion of *dataspaces* aims at providing the benefits of the classical data integration approach, but in a progressive way [29, 75, 34]. In short, a *dataspaces* environment does not focus on offering an ideal, integrated view of the resources, but the tools and mechanisms that allow users to execute integration cycles if benefits worth the required effort.

In a data integration spectrum, the classical approaches are at the high-cost/high-quality end, while incremental integration based on small steps appears on the opposite side. The *dataspace* is continuously refined to improve the connections among sources, being divided between a bootstrapping stage and subsequent refinements. Improvements within a *dataspace* can be based, for instance, on structural analysis [22], relevance-feedback [6] or on manual/automatic mappings among sources [35].

## 4.4 LinkedScales

Progressive integration proposals have brought contributions in domains like private: information management [14, 19], Web-related artifacts [53], and justice [77]. However,

related work lacks an architecture that systematizes the progressive integration process, keeping tracking of transformations and allowing reuse of intermediary results in multiple applications [34].

*LinkedScales* is a framework, developed by us, which defines a multiscale graph-based data model and an architecture built over an instance of the model. The three fundamental pillars of *LinkedScales* are: systematizing integration steps [60]; allowing provenance between steps; and supporting the reuse of partial results [56]. To achieve such goals, *LinkedScales* organizes the integration steps as a stack of scales (subgraphs within the dataspace), where objects of an upper scale are built based on transformations over objects of a lower scale.

Scale transformations within the dataspace are tracked by an orthogonal graph, supporting traceability among tasks by correlating sources and targets to transformations between scales. The model supports scale-specialized annotations, named trails, that enable users to refine the dataspace on demand. Such aspects are also discussed in previous publications [56, 57, 60, 61].

In this paper, we present our multiscale architecture applied to the health domain. It involved the design of a conceptual model for the medical training context (introduced in Section 4.2), as well as, the definition of a transformation plan encompassing the whole stack. Some aspects of the data model were extended and refined, and are presented here. Furthermore, we present an exploratory analysis regarding how the provenance data of the orthogonal graph can foster dynamic adaptations of scales according to constraints.

#### 4.4.1 The Data Integration Architecture

The *LinkedScales Primary Integration Architecture* defines a mandatory, systematic initial set of scales. They are based on architectural patterns faced by us in previous experiences [59, 8, 55], and aims at emphasizing the different levels of integration and their particular abstractions. Further domain or application related scales can be derived on top of the Integration Architecture.

Fig. 4.5 presents the *LinkedScales Primary Integration Architecture*. It shows its four mandatory scales, going from the raw data sources (lower scales, containing more format and structural details) towards a conceptual scale (less details of format and structure, more focus on concepts).

Integration in the architecture starts with the ingestion of data sources to our dataspace, representing them as graphs, on the lowest scale. This process is a bootstrapping effort and aims at homogenizing the access to resources, taking advantage of the flexibility and low footprint of graphs to logically represent different types of structures.

Each subsequent scale is derived from the previous one. Since resources are always graphs, transformations within and across scales are reduced to graph operations. Graphs in the architecture follow patterns according to the scale. Inspired by the layered architectural style, each algorithm of a given scale can rely on expected characteristics of data of the respective lower scale and focus on its duties. Therefore, the algorithms become portable and reusable in different applications – i.e., different dataspace setups – as long as they work in their native scale. This strategy enables encapsulating in reusable modules

results of efforts to solve specific challenges.

We further summarize the role of the scales from bottom to top, as shown in Fig. 4.5. Find further details in [56].

The role of the **Physical Scale** is to homogenize the physical representations via graphs, making structures of the original resource explicit and homogeneously linkable. A Graph Translator [59, 56] reads several specialized formats – *e.g.*, Excel, CSV, Relational tables, XML – and converts them to an equivalent graph representation.

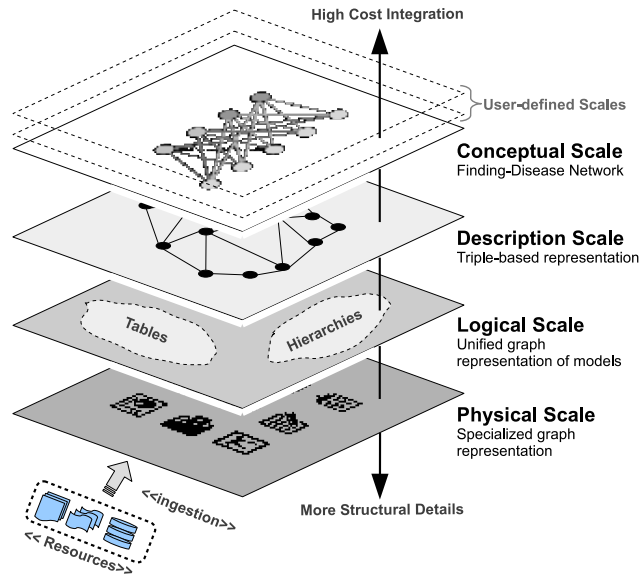


Figure 4.5: LinkedScales Primary Data Architecture applied to the EBM Scenario.

The benefit of having a **Logical Scale** is to provide a standard view over data following similar or equivalent logical models. Tables and hierarchical documents are examples of logical models present in the resources containing EBM data.

Files have specificities in their formats. While, on the Physical Scale, differences will exist when representing – *e.g.*, a table within a PDF, a sheet from a spreadsheet or a table within an HTML page – on the Logical Scale, format specificities disappear, and logical models are represented following the same structural standard – *e.g.*, all tables will follow the same logical model, in spite of their origin.

Differently from the Logical Scale, the **Description Scale** emphasizes the content – *e.g.*, values in spreadsheet cells – and their relationships, minimizing the structural differences.

Models represent relations among data elements in different ways – *e.g.*, a row in a table can represent data concerning the same entity, while hierarchical relationships in a document represent aggregations of textual elements. The Description Scale comprises efforts on mapping all logical models to a single unified one, shifting the focus towards the content. This scale relies on the descriptive model based on the triple <resource, property, value>, which is a convention in several metadata standards as *Resource Description Framework* (RDF).

Besides the model unification, fundamental semantic aspects of RDF are not present



on this scale yet. It is the main role of the next scale, which implies discriminating entities, adopting controlled vocabularies to represent descriptive properties, and explicitly expressing the semantics of elements via ontologies.

The **Conceptual Scale** aims at wrapping up the efforts undertaken in previous scales and representing the dataspace on a semantic level. It uses the content and relationships between the elements of the previous descriptive graph to discover and make explicit the semantics via ontologies. Entities are discovered, deduplicated and related to ontologies as instances of classes or properties. The simple, “textual graph” of the previous scale becomes a semantic graph containing interrelated entities and their properties/values, with explicit meaning supported by predefined ontologies.

#### 4.4.2 The underlying multiscale graph-based data model

This section briefly resumes the elements of the LinkedScales underlying graph data model, defined in previous work [56]. The elements introduced here are the basis of the model extension presented in Section 4.4.3.

Likewise the model of *graph databases* [83, 3], LinkedScales is a finite, edge-labeled, directed graph. Let  $\Sigma$  be a finite alphabet and  $\mathcal{V}$  a countably infinite set of ids. A LinkedScales  $L$  over  $\Sigma$  is a tuple  $(V, E, F)$ , being  $V$  a finite set of **nodes** and  $E$  a finite set of **edges**, where  $V \subseteq \mathcal{V}$  and  $E \subseteq V \times \Sigma \times V$ .  $F$  is a function  $F : V \rightarrow \Sigma$ , which associates a vertex to a label.

Given any two **scales** within the dataspace,  $S_i = (V_i, E_i)$  and  $S_j = (V_j, E_j)$ , where  $V_i \subseteq V$  and  $V_j \subseteq V$ ,  $V_i \cap V_j = \emptyset$ . Given two nodes  $u, v \in V$  and a label  $a \in \Sigma$ , an edge  $e \in E$  is a triple  $(u, a, v)$  indicating a link between  $u$  and  $v$  with a label  $a$ . **Paths** within a scale are a set of edges in  $E$  connecting two nodes (initial and final) – i.e.,  $\pi = \{(v_1, a_1, v_2), (v_2, a_2, v_3), \dots, (v_{m-1}, a_{m-1}, v_m)\}$ , where any edge  $(v_{i-1}, a_{i-1}, v_i) \in E$ .

**Objects** are units within a scale of the dataspace, which are the atomic units of any transformation between scales. An object  $O_h$  belongs to a scale  $S_i$  if all nodes/edges of the paths in  $O_h$  are nodes/edges of  $S_i$ . Therefore, any transformation between two scales is defined in terms of transformations between objects of the respective scales. An object is defined as a set of paths  $O = \{\pi_1, \pi_2, \dots, \pi_r\}$ .

**Trails** are structured annotations that guide transformations within the dataspace – as will be further discussed in Section 4.5, trails can be manually inserted or automatically inferred. In this model, a trail  $t_x = (O_x, a_y, n_z)$  connects an object  $O_x$  and a node  $(n_z)$  via an edge with a label  $a_y$ .

**Transforming** a lower scale to an upper scale can be reduced to a two-step process: match and transform. The *match* step aims at fetching objects in the subgraphs of a given scale, while the *transform* step addresses the production of a transformed subgraph within the upper scale. Intermediary processing and scale related steps are performed during the execution of match-transform operations. Match and transform operations are encapsulated in the concept of criterion. A **criterium**  $\mathcal{C}_\alpha$  is a set of criterion  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ . A **criterion**  $\lambda_a$  is a pair  $(m_a, t_a)$ , where  $m_a$  is a **match** operation and  $t_a$  is a **transform** operation.

**LinkedScales**, therefore, is a tuple  $\mathcal{LS} = (S_i, \Omega, \mathcal{F}_{st})$ , where  $S_i$  is the first scale,

$\Omega = \{C_1, C_2, \dots, C_n\}$  is a sequence of transformation criteria, and  $\mathcal{F}_{st}$  is a function  $\mathcal{F}_{st} : S_i \rightarrow S_{i+1}$ , which derives a upper scale  $S_{i+1}$  by applying a transformation criteria,  $C_i$  over a scale  $S_i$ .

#### 4.4.3 Extending the data model

Based on the model defined in a previous work [56] and briefly introduced in the previous section, in this paper we present a refinement and extension of the model to define: (i) how paths/objects are fetched within scales, (ii) how the orthogonal transformation graph track interscale transformations, and (iii) how trails can be attached to scales and be contemplated in the transformation graph.

##### Fetching paths

A **match** operation  $m_a$  – which is part of a **criterion**  $\lambda_a$  – aims at fetching one or more objects within a scale. Objects, as set of paths, are realizations of regular path queries. A **regular path query** (RPQ) is a basic querying mechanism of graph databases [15, 17, 3] and is applied here over a specific scale as part of a match sentence.

RPQs focus on finding pairs of nodes connected by a path, in which labels belong to a given regular expression. More formally, a RPQ is a tuple  $R_q = (n_a, L, n_b)$ , where  $L \in \Sigma$ , is a pattern to represent potential paths as a regular expression [3]. Therefore, given a scale  $S_i = (V_i, E_i)$  and a RPQ  $R_q = (n_a, L, n_b)$ , the expected result for  $R_q(S_i)$  is the set of all possible paths connecting  $n_a$  and  $n_b$  that matches the expression  $L$ .

There are multiple languages for querying graph data models [2, 4]. Each language and each data model has particular features for representing graph patterns according to the expected application of the language. These features operatically impact or limit different categories of queries [3, 39, 41].

To implement our model, we selected a largely adopted query language named Cypher<sup>1</sup> as the underlying basis for defining patterns aligned with our model. Therefore, we further use the graph operators from Cypher to represent patterns.

The declarative, SQL-inspired characteristics of Cypher allow the description of patterns in graphs via an ascii-art syntax [39]. Consider the following structure:

**(node1:Label1) - [edge1:Label2] -> (node2:Label3)**

In Cypher, nodes are surrounded with parentheses, e.g.,  $(node)$ . Edges/relationships assume a arrow-like structure  $(-->)$  connecting two nodes and additional information and regular expression operators can be placed in square brackets inside of the arrow. The query below is an example of how to write a pattern for retrieving paths between two nodes, using the  $*$  wildcard – e.g., the wildcard symbolize the repetition of sub-patterns. In the LinkedScales data model, Cypher-based patterns are adopted for retrieving paths.

**MATCH path = (node1:Label1) - [ \* ] -> (node2:Label2)**  
**RETURN path;**

---

<sup>1</sup><http://neo4j.com/docs/stable/cypher-query-lang.html>

Consider, for instance, the scenario illustrated in Figure 4.6. It represents an abstraction of a table ( $table_1$ ) and a hierarchy ( $hierarchy_1$ ) within a scale ( $Scale_i$ ). It also highlights (dashed line) an object ( $object_1$ ) and a path ( $path_1$ ).

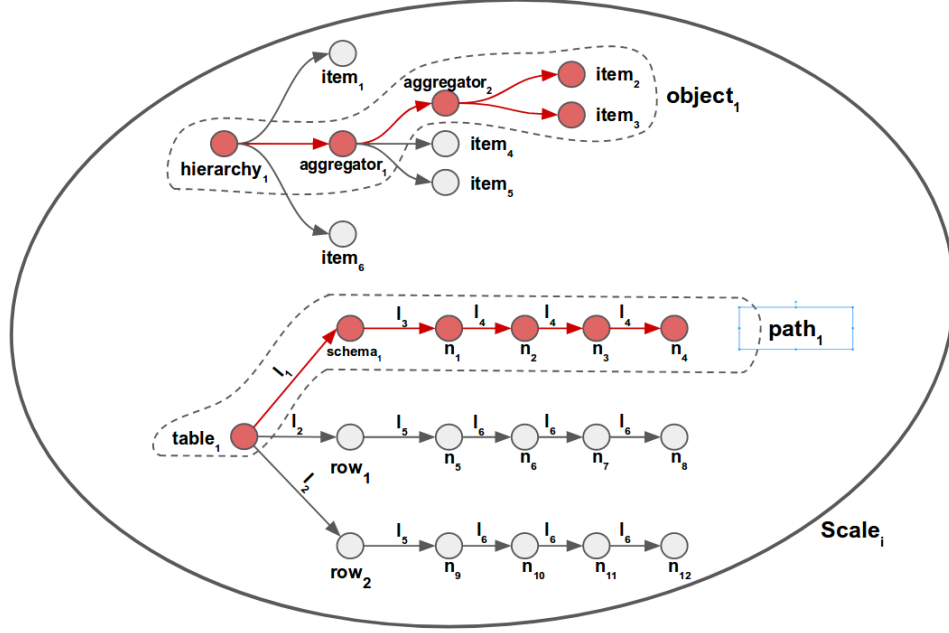


Figure 4.6: Example of a path and object within a scale.

The  $path_1$  in Figure 4.6 connects node ( $table_1$ ) to node ( $n_4$ ). Such path comprises the node that represents the table and a chain of nodes of the table schema. The RPQ for retrieving  $path_1$  can be defined as  $R_{q1} = (table_1, E_1, n_4)$  in Cypher. The expression  $R_{q1}$  can be adapted to a Cypher query as:

```
//IN Scale_i :

MATCH path_1 = (:table {id:1})-[:l_1]->(:schema)-[*]->(:n {id:4})

RETURN path_1;
```

Code 4.1: Cypher query pattern for retrieving  $path_1$  of Figure 4.6

In this expression, words preceded by a colon are Cypher labels. Curly brackets define pairs of property/value related to nodes or edges. Indexed labels adopted in our example of Figure 4.6 were mapped to a combination of a Cypher label plus a property  $id$ , whose value is the index.

$object_1$  of Figure 4.6 comprises a set of two paths. The first path connects  $hierarchy_1$  to  $item_2$  and the second connects  $hierarchy_1$  to  $item_3$ . The RPQ for the first path is, therefore,  $R_{q2} = (hierarchy_1, E_2, item_2)$  and for the second one is  $R_{q3} = (hierarchy_1, E_2, item_3)$ . Even though Cypher allows simpler patterns to reach both paths in only one query, we

represent here a construction for each path. The expression  $R_{q2}$  adapted to Cypher becomes:

```
//IN Scalei :

MATCH path2 = (:hierarchy {id:1})-->(:aggregator {id:1})-->
              (:aggregator {id:2})-->(:item {id:2})

RETURN path2;
```

Code 4.2: Cypher query pattern for retrieving one path of  $object_1$  of Figure 4.6

Similarly, the expression  $R_{q3}$  mapped to a Cypher query is:

```
//IN Scalei :

MATCH path3 = (:hierarchy {id:1})-->(:aggregator {id:1})-->
              (:aggregator {id:2})-->(:item {id:3})

RETURN path3;
```

Code 4.3: Cypher query pattern for retrieving the second path of  $object_1$  of Figure 4.6

The set of vertices and edges returned by  $path_2$  and  $path_3$  of previous Cypher queries is the same of  $object_1$  of Figure 4.6. Therefore,  $object_1$  can also be defined as  $object_1 = (R_{q2}(Scale_i), R_{q3}(Scale_i))$ .

Objects delimit a portion of a scale which is subject to a transformation, enabling traceability, i.e., it is possible to track which portion of a scale produced which portion of another scale. Being a set of paths, objects can be defined and retrieved via Cypher-based queries.

## Inter-scale Transformations and the transformation graph

Successive transformations permeate the construction of the highest scale within a Linked-Scales dataspace. Each transformation may involve systematic match and transform operations (criteria), fetching objects – atomic transformation unities – from a lower scale to produce objects in an upper scale.

*LinkedScales* produces a provenance graph while transformations are executed. This graph contains not only information about processes, but also which operational annotations (trails) were considered for the transformation.

Objects of a given lower scale are subgraphs defined by the match clause of a criterion, as well as objects of the respective upper scale are derived subgraphs built according to a transform operation. The orthogonal graph is named Multiscale Transformation Graph

(MTG) and is disjoint from the data subgraphs (scales). The traceability provided by the MTG fosters future analysis of provenance, reproducibility, *etc.*

Figure 4.7 depicts how the MTG is represented within LinkedScales, connecting objects ( $object_1$  and  $object_2$ ) from  $Scale_i$  to  $object_3$  from  $Scale_{i+1}$ . The MTG format is based on the *PROV Ontology* (PROV-O) [49]. PROV-O *Entities* are objects, trails and other resources used during transformations, while the transformation executed ( $task_1$  of the figure) is a PROV-O *Activity*. Further details on the MTG are also discussed in [56].

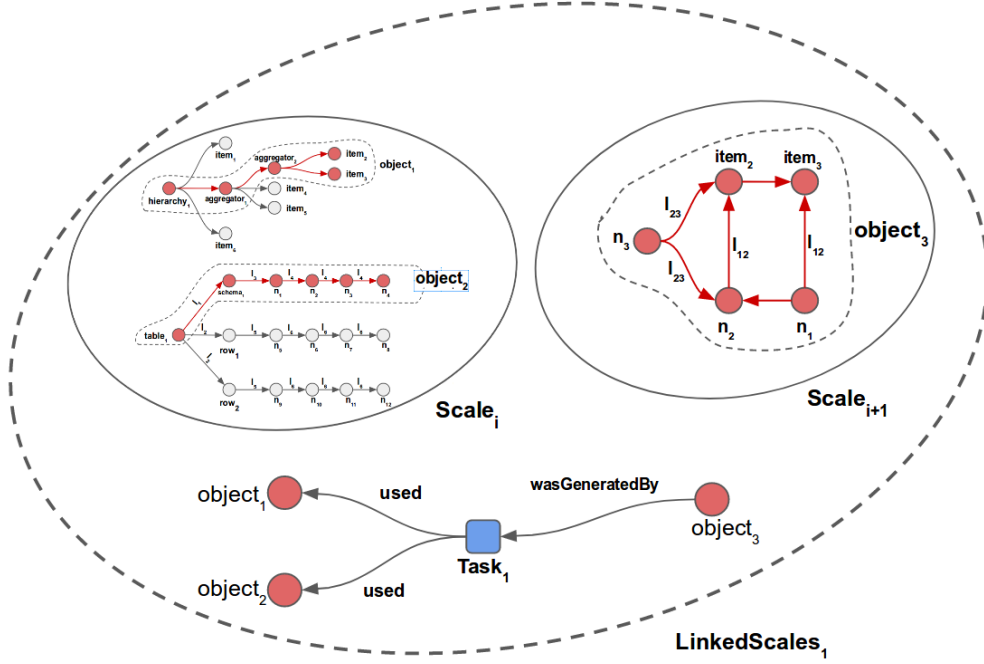


Figure 4.7: Example of transformation: two objects as input, a third object as output

## 4.5 Building a Clinical Feature-Disease Dataspace

Several initiatives are gathering and making available evidence-based clinical data that can empower better point-of-care decisions. However, such initiatives usually focus only in organizing [47], synthesizing [25], and indexing [71, 38, 11] research publications. Therefore, EBM information is still scattered along multiple repositories in different formats and schemas, hampering the construction of a unified view of data.

Current strategies for integrating EBM data often result in the development of specialized solutions or in an intensive manual task of trimming and organizing portions of data [11]. In this section, we describe how the LinkedScales stack (discussed in the previous section) has been applied as the underlying foundation to integrate EBM data from multiple resources and to produce a semantic model interconnecting evidence. We further describe the experimental scenario, implementation aspects, the bootstrapping transformations and how manual annotations (trails) are used to refine data within the dataspace.

### 4.5.1 Experimental Scenario

Figure 4.8 illustrates the goal of our experiment. It shows an example of how data within an EBM paper (table in the right side) can be extracted to build a network of associations among data elements from the table (red subgraph) and concepts from the *Heart Failure Ontology* (HFO) (blue subgraph). The EBM paper is a meta-analysis relating clinical history, physical examination, and symptoms with the diagnosis of acute aortic dissection [44].

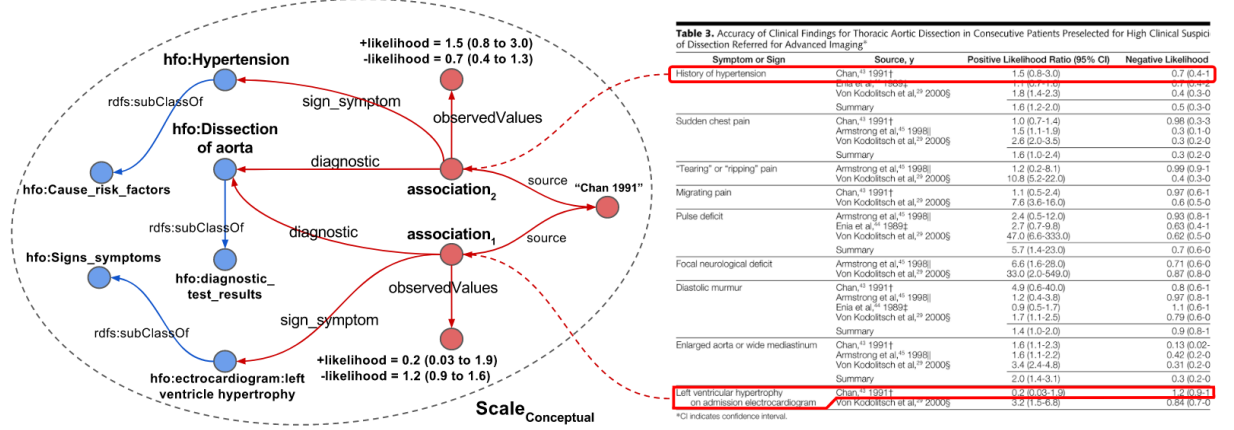


Figure 4.8: Connecting elements from Heart Failure Ontology with data extracted from a meta-analysis paper (table from [44])

Following the same procedure, physicians from the Emergency Medicine Department – Faculty of Medical Sciences, University of Campinas – selected well-conducted research papers, which have been ingested and integrated into a Clinical Feature-Disease network on the conceptual scale. They were related to the following diseases: Tension Pneumothorax; Pulmonary Embolism [16, 43]; Esophageal Perforation & Rupture; Acute Myocardial Infarction [64]; and Aortic Dissection [44].

Our focus was the five life-threatening conditions that must be readily confirmed or excluded when physicians are diagnosing patients reporting chest pain as the chief complaint in the context of emergency rooms [9, 68] – as discussed in Section 4.3. Although the scenario described here focuses on papers regarding specific diseases, the strategy can be extended to other scenarios and further resource types.

The example depicted in Figure 4.8 will drive the further discussion on how the EBM data is integrated within the LinkedScales dataspace. Subsequent sections show step-by-step how the selected sources were transformed from raw representations (physical scale), to homogeneous representations (logical scale), descriptive representations (description scale), and finally to the conceptual representation. Furthermore, challenges on tracking uncertainty during the scale transformations are discussed.

### 4.5.2 Implementation Aspects

The LinkedScales System Architecture is illustrated in Figure 4.9. The system was built on top of the Neo4j graph database and further modules were implemented as web services. To import resources as graphs in the *Physical Scale*, we adopted a framework, developed by us, called *2graph*<sup>2</sup>. The framework, introduced in [56], is able to convert documents (DOC, DOCX, and some PDFs), tables within HTML pages, XML, and spreadsheets (CSV, ODS, XLS, XLSX) as graphs in either Neo4j (property graph) or Virtuoso (triple store).

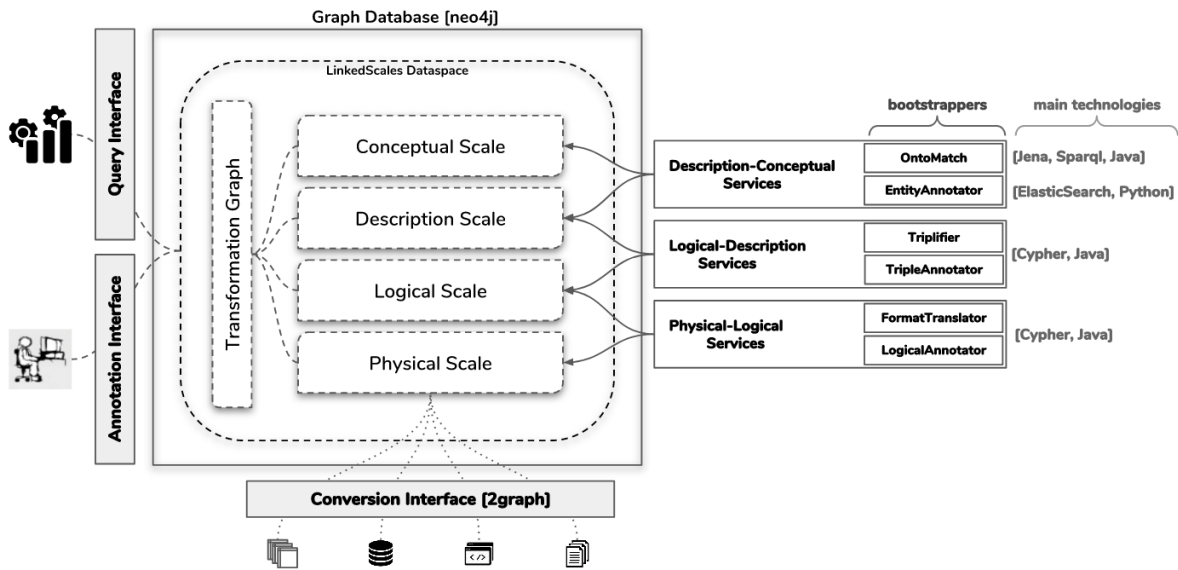


Figure 4.9: Overview of the implemented LinkedScales architecture

As shown in the right part of Figure 4.9, between any two scales, there is a set of services divided into two main roles: bootstrap annotate and transform. The former automatically produces trails on a scale to guide transformations to its respective upper scale. Trails can be further manually refined and corrected. The latter applies the transformation driven by trails.

These roles are performed respectively by the *LogicalAnnotator* and *FormatTranslator* services between the physical and logical scales. The *LogicalAnnotator* service is specialized in recognizing logical formats inside each specific internal representation. It adds trails to the physical representation indicating, for instance, the table structure of an ingested CSV file, including which fields correspond to the schema. Based on these trails, the *FormatTranslator* service converts different physical formats in a homogeneous logical format.

By the same token, while the *TripleAnnotator* service recognizes and annotates RDF-like ( $\langle resource, property, value \rangle$ ) triples in the several logical formats, the *Triplifier* service produces the respective triples in the *Description Scale*. An example of such transformation is discussed in Section 4.5.3.

<sup>2</sup>Available at <https://github.com/matheusmota/2graph>

The same roles are performed by the *EntityAnnotator* and *OntoMatch* services respectively to produce the Conceptual Scale. They are detailed in the next sections.

### 4.5.3 From sources to the Conceptual Scale

Figure 4.10 illustrates the sequence of transformations from the *Physical* to the *Conceptual* scale. In the Physical Scale, although we show the original PDF table, we are considering its corresponding graph.

The transformation depicted in Figure 4.10 from box [A] to box [B] illustrates how the raw format of a table within a PDF – containing specific metadata concerning the position, format *etc.* and no explicit schema – is transformed to a graph in the *Logical* Scale, following a common *Table* structure, i.e., there are nodes to delimit a table, its schema and rows.

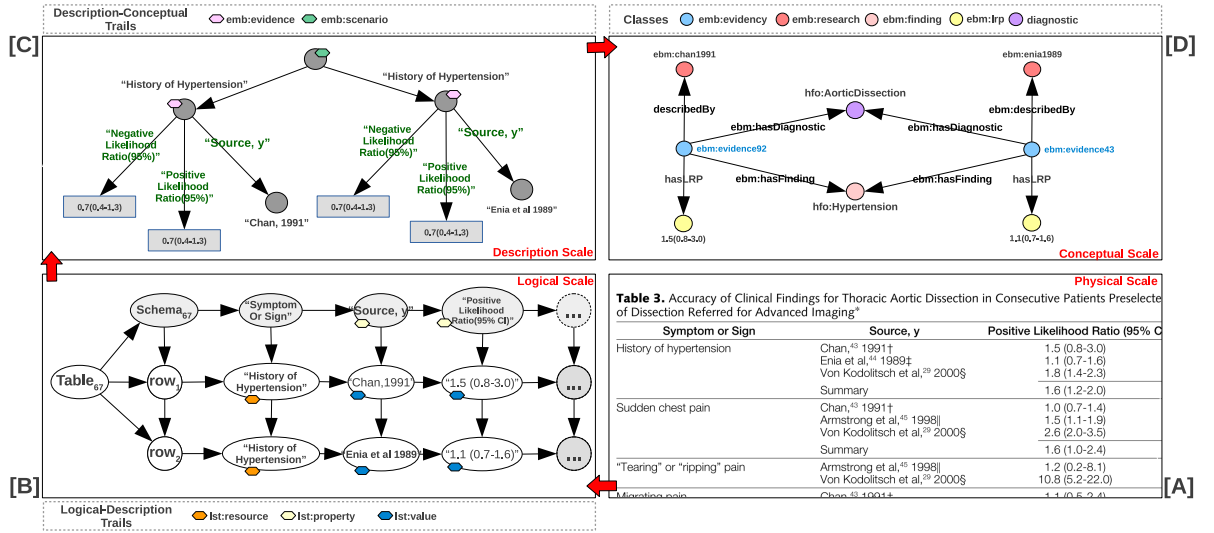


Figure 4.10: Example showing the sequence of transformations from the Physical to the Conceptual scale

Figure 4.10 [B] shows trails produced by the *TripleAnnotator* service, pictured as small colored hexagons over the nodes. Orange, yellow and blue trails indicate elements inside the table to be transformed into resource, property, and value respectively. An advantage of the effort to homogenize various formats behind the same logical model is the chance of reusing algorithms over the same logical structure, independently of its physical format – *e.g.*, the same *TripleAnnotator* will work over any table without concerns of specific formats.

Based on the trails, the *Triplifier* service transforms rows (nodes  $row_1$  and  $row_2$ ) in resources, schema attributes (green nodes "Symptom or Sign", "Source", "Positive Likelihood Ratio (95% CI)") in properties and cells in values. It reduces this and other logical models to an RDF-based model. However, it still not a full-fledged semantic graph – i.e., the content of the nodes and edges are still plain text and are not proper to be interpreted by machines.



The *EntityAnnotator* service recognizes entities and annotates them. Figure 4.10 [C] illustrates trails designating evidences. In our example, such annotations aim at making the semantic of EBM-related elements explicit, by adopting specific vocabularies. The next section details how the *OntoMatch* service produces the *Conceptual Scale* considering these trails. This specific transformation is emphasized here since Physical-Logical and Logical-Description transformation processes were detailed in previous publications.

Algorithms adopted for transformation – e.g., table or hierarchy "*triplification*", schema recognition, entity linking, etc. are widely discussed in the literature (including a previous work developed by us [8]) and are not a subject of attention in this research.

## Building the Conceptual Scale

Figure 4.11 expands the description-conceptual transformation shown in Figure 4.10, detailing its criterion – i.e., match and transform operations as described in Section 4.4.2. It aims at representing EBM-related data of the description scale as an ontology-like structure in the conceptual scale. As described in Section 4.4.3, the pattern in the *match* operation is defined by a regular expression.

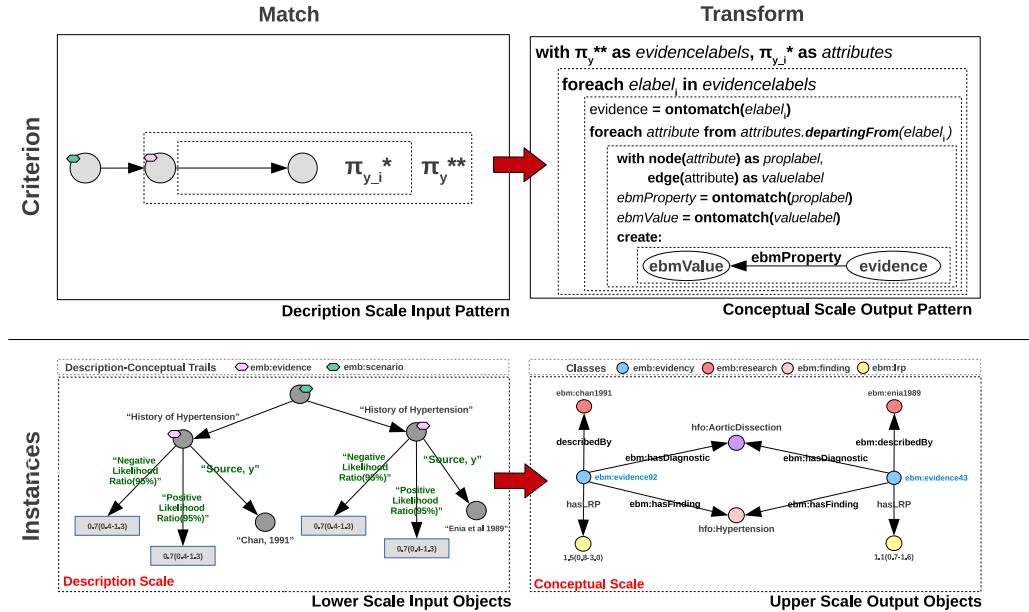


Figure 4.11: Example of a criterion transforming data from the description to the conceptual scale

The figure adopts a diagram to express nested regular expressions, which can be mapped to the model presented in Section 4.4.3 through nested variables, i.e., the inner regular expression is assigned to a variable that appears in the outer regular expression.

In this diagram, wildcards symbolize the repetition of sub-patterns in different scopes. The expression  $\pi_{y,i}^*$  indicates the repetition of the edge/node pattern inside the dashed box. All subgraphs of this scope must be connected to the node of the outer scope, with an *ebm:evidence* trail, represented by a pink hexagon. It acts as an aggregate node. Considering that this match expression is applied to the *Description Scale*, each subgraph that matches the expression at this point will be a resource (evidence) and all

related property/values. The second wildcard `**` means a repetition in the outer scope. Following the same rationale, it represents a set of evidence nodes connected to the same *ebm:scenario* (node with the green trail).

The *Transform* part of Figure 4.11 is defined by a pseudocode similar to the Cypher query language<sup>3</sup>. The *"with"* clause defines a scope and the respective variables, comprising the set of instances met and returned by the match pattern. The clause *"with  $\pi_y^{**}$  as evidencelabels"*, for instance, implies that all paths matching the pattern  $\pi_y^{**}$  will be available in the scope inside the clause through the variable *evidencelabels*. Inner *"foreach"* clauses iterate along the variables produced in the outer scope. The *"ontoMatch()"* function triggers the *OntoMatch* service, further detailed.

#### 4.5.4 Entity Resolution in the Conceptual Scale

The last stage of the integration involves the entity resolution process, i.e., unequivocally identifying entities from textual descriptions. In our case, they are related to concepts in ontologies. The *Heart Failure Ontology* (HFO) [42, 80] was adopted as the primary ontology in this experiment since most of the selected diseases are related to heart problems. It contains 1,652 classes concerning heart failure relevant information. Concepts from the *Clinical Signs and Symptoms Ontology*<sup>4</sup> (CSSO, 303 classes) and the *Symptom Ontology*<sup>5</sup> (SYMP, 942 classes) were considered during the experiment as well.

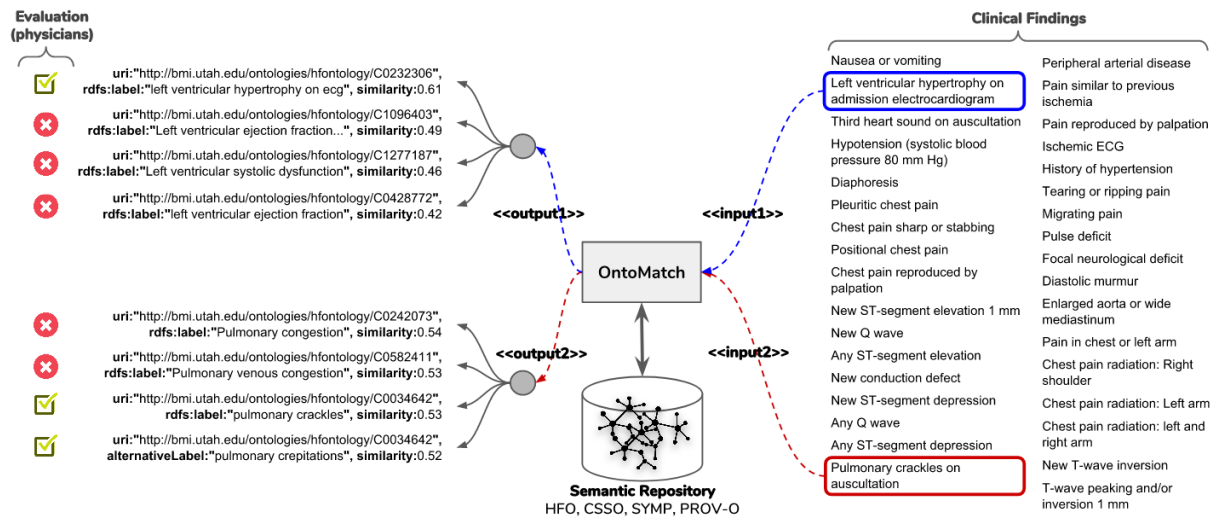


Figure 4.12: Example of Clinical Findings and related concepts found by the OntoMatch service.

There are multiple strategies for resolving entities, varying from string edit distance to more specialized hierarchical analysis of record attributes [27, 12]. We have used OntoMatch<sup>6</sup> as a bootstrapping transformation service to resolve entities in the dataspace.

<sup>3</sup><http://neo4j.com/docs/stable/cypher-query-lang.html>

<sup>4</sup>Available at <http://mdb.bio.titech.ac.jp/csso>

<sup>5</sup>Available at <http://symptomontologywiki.igs.umaryland.edu>

<sup>6</sup>Service available at <http://ontomatch.lis.ic.unicamp.br> and source available at <http://github.com/faguim/ontomatch>

The service parses a given text and returns associated entities, identified by URIs, within an ontology. Even though the service was designed to afford different strategies, the current implementation supports distance measures based on string similarity, applied to the *rdfs:label* property and related annotations (*owl:AnnotationProperty*) – e.g., OWL alternative labels – of each entity.

Figure 4.12 shows the resulting of an OntoMatch execution. On the right, there are clinical findings (evidence) textually described, matched from the *Description Scale*. For each clinical finding, the OntoMach service produces a set of ranked candidates, displayed on the left side. For example, the “Left ventricular hypertrophy on admission electrocardiogram” clinical finding was related to four concepts (left top). Each association has a related similarity value, in this case, the distance of the involved strings.

This automatic bootstrap association can be further refined by physicians. For example, the two associations displayed in Figure 4.12 were submitted to evaluation of physicians. For the four candidates for the “Left ventricular hypertrophy on admission electrocardiogram” clinical finding, only the first was considered equivalent; for the “Pulmonary crackles on auscultation”, only the third and fourth. The automatic associations and the further refinements are all captured by our Multiscale Transformation Graph (MTG) (see Section 4.4.3), as detailed in the next subsection.

#### 4.5.5 Multiscale Transformation Graph and Uncertainty

Figure 4.13 shows the graphs that correspond to the association of the “Left ventricular hypertrophy on admission electrocardiogram” clinical finding to the respective ontology class, as introduced in the previous section. Besides the paths involved in the *Description* and *Conceptual* scales, the MTG is also detailed.

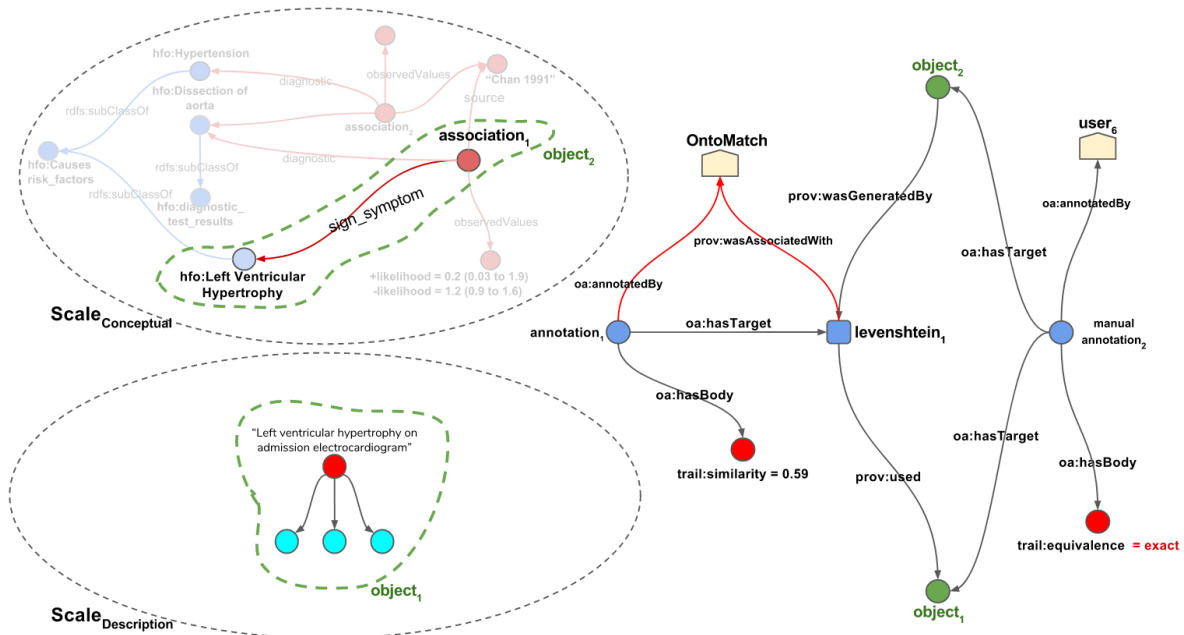


Figure 4.13: An example illustrating an association of a clinical evidence to an entity and the respective MTG.

The figure shows a path in the *Description Scale* that was selected based on a Match expression. As described in Section 4.4.3, the path is connected to an object (Object 1). Each object in this scale fetches a clinical finding.

The *OntoMatch* service associates the clinical finding to the ontology class, producing the Object 2 of the *Conceptual Layer*. The association is recorded in the MTG following the PROV-O standard, i.e., the blue square node (*prov:Activity*) indicates the activity that produced the association between the source (*prov:used*) and the target (*prov:wasGeneratedBy*). It is related with the *OntoMatch* service (*prov:Agent*), who performed the task. There is an annotation connected to the activity – follows the Open Annotation Data Model [72] – indicating the similarity found in the strings. When a physician confirms the association, as described in the previous subsection, it produces a second annotation.

Both annotations are fundamental to keep track of the quality and confidence level of the produced information. The MTG can be explored, for example, to filter information according to a certain level of similarity.

The used algorithms for evaluating string similarity and distance measures via the *OntoMatch* service are the following: Levenshtein [51, 79], Normalized Levenshtein<sup>7</sup>, and Jaro-Winkler [82]. Furthermore, distance measuring strategies such as *Cosine Similarity* and *Jaccard index* are available [63].

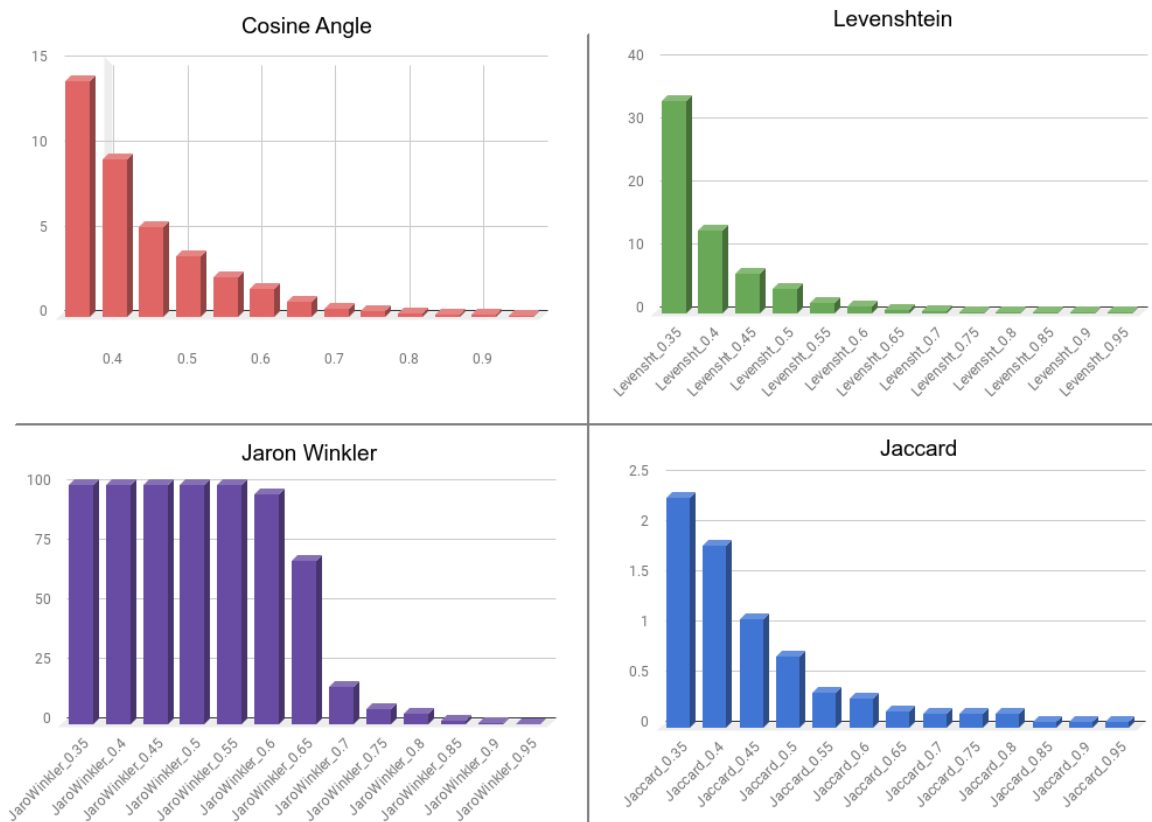


Figure 4.14: Relation between the average number of candidate concepts per Clinical Finding and the minimum similarity threshold.

<sup>7</sup>Levenshtein distance divided by the length of the longest string, resulting in a value in the interval [0.0 1.0]

The described annotation process was tested with three physicians who collaborate in this project. The bootstrapping OntoMatch service ranked similar concepts associated to the 39 distinct Clinical Findings extracted from the ingested research papers, as detailed in our Experimental Scenario (Section 4.5.1).

The service looks for concepts in the ontology that have the highest similarity ( $similarity \in [0, 1]$ ) with a given string. For each algorithm, Figure 4.14 shows the relation between the average number of candidate concepts per Clinical Finding and the minimum similarity threshold (varying from 3.5 to 0.9). Loose similarity thresholds implicate in more candidates per Clinical Finding.

In order to select the proper candidates to be annotated by the physicians, the MTG was exploited as shows Figure 4.15. The figure shows three stages of a simplified representation of the transformation depicted in Figure 4.13, where nodes of the *Description Scale* are associated to nodes of the *Conceptual Scale*, connected by the MTG recording similarity value. In Stage 1 a high value for the accepted similarity threshold produces a large number of candidates, most of them poorly related. The MTG enabled us to adjust this similarity threshold, filtering minor distances without losing too many relevant options, reaching Stage 2.

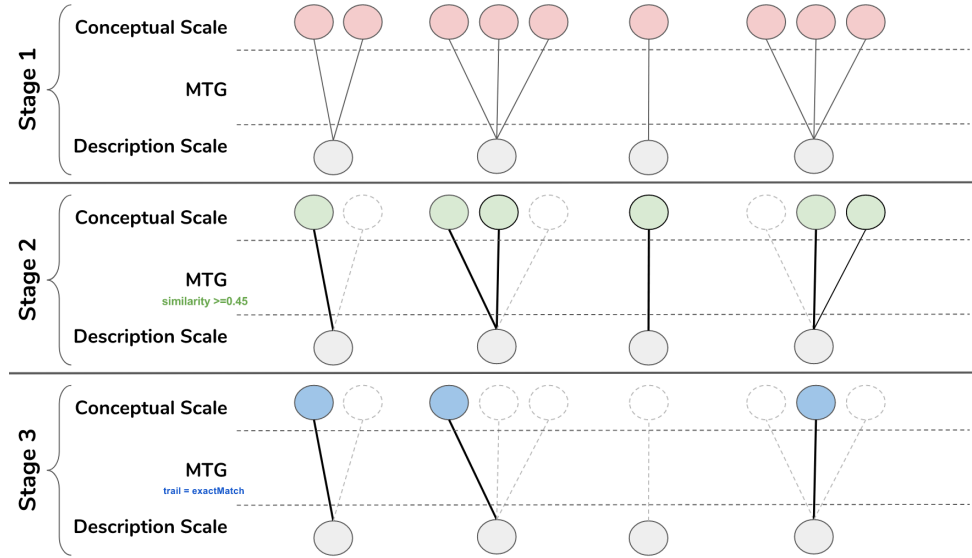


Figure 4.15: Filtering and annotating evidences according to similarity.

This is a visual interactive process that we performed, but that can be conducted by specialists (physicians) in the future. Through this process, we chose a threshold of 0.45, producing an average of 5.7 candidates for each Clinical Finding (1.7 of standard deviation). The result was presented to the physicians to be annotated through a form, with the 38 clinical findings and 216 candidate ontology entities. The result was 13 clinical findings with exact match, 47 with partial match, and 156 classified as different. These exact matches were annotated in the MTG (Stage 3) and appear in our navigation interface as described in the next section.

### 4.5.6 Navigating on the Conceptual Scale

The produced *Conceptual Scale* depicted in Figure 4.10 [D] feeds a querying interface<sup>8</sup> for navigating along the graph. Such interface allows users to see the interaction of clinical features and the life-threatening diseases as mentioned before.

Figure 4.16 presents a *screenshot* of the interface prototype, showing the data corresponding to the example discussed in Section 4.2 with a pretest of 25%. The interface allows the user to select observed clinical findings and shows what will be post-test probability for each possible diagnosis. The selected clinical finding (*Chest pain radiation: both arms*) implies in a 70.3% post-test probability for Acute Myocardial Infarction since the likelihood ratio is 7. An interesting feature of the interface is showing the source papers from which the data were extracted to relate the clinical findings to the diagnosis. To fetch these papers, the interface explores the orthogonal graph (MTG), tracing the operations from the clinical findings to the original papers.

The screenshot displays the EBM-DB interface for a 'Thoracic Pain' scenario. The 'Scenario Setup' section includes a context of 'Emergency', initial clinical features of '[Male][Age > 65y]', and a pretest of 25%. Under 'Clinical Findings', 'Chest Pain' and 'Chest pain radiation: Both Arms' are selected. The 'Diagnostic Analysis' section shows posttest probabilities for several conditions: Aortic Dissection, Esophageal Rupture, Pulmonary Embolism, Acute Myocardial Infarction (70.3%), Tension Pneumothorax, and Costochondritis. The 'Acute Myocardial Infarction' result is highlighted with a red bar and labeled 'life-threatening'. The interface also includes a sidebar with navigation options and a section for 'Associated EBM Literature' showing one paper from 'Jama'.

Figure 4.16: Screenshot of the prototype for navigating in the conceptual scale.

<sup>8</sup>Demo and data available at: <http://ebm.lis.ic.unicamp.br/>

## 4.6 Conclusion

Medical training has been challenging the health community, especially in front of the ever growing available knowledge. The extent source of systematized knowledge, produced by evidence-based medicine, can be explored to support medical training by computational tools. In our context, we showed how we are working to build a unified knowledge base compiling data scattered in several sources.

We presented in this paper how we addressed the challenge applying our multiscale dataspace architecture, factoring different aspects of the problem per scale, enabling to reuse specific algorithms aimed to solve specific issues.

The focus of this paper was in the dataspace construction and the validation of the transformation process, from raw data to the conceptual model. The game that fully exploits this dataspace is a work in progress. Future work also includes the refinement and formalization of a method to assess and guide the student comparing the expected and executed paths in the graph.

## Chapter 5

# Conclusions and Future Work

This thesis presented and discussed a graph-based model and system architecture for a dataspace. It systematizes in layers (scales) progressive integration steps, based on graph transformations. The model is founded in previous work, which explored different aspects of the proposal.

LinkedScales is aligned with the modern perspective of treating several heterogeneous data sources as parts of the same dataspace, addressing integration issues in progressive steps, triggered on demand. We have designed a generic architecture able to be extended to several contexts.

The work combined theoretical contributions, in the form of an abstract model, with the design and implementation of an architecture. The practical applications in two distinct domains (biology and health) were important to test the generality of our proposal and to better validate it. It is also aligned with the recent evolution of graph databases and their applications. The flexibility provided by graph databases and their lightweight schemas, which foster dynamic adaptations and exploratory analysis, are aligned with our architecture and its pay-as-you-go integration approach.

Besides the practical result of an architecture, the work impelled us to produce an abstract model, contemplating characteristics that can be exploited beyond LinkedScales. Our approach to delimiting objects in graphs – in order to relate them and track transformations – can be applied, for example, in version control.

One important aspect of compartmentalize the steps of the work in multiple scales, which was not explored in this work, is the possibility of collaboration of many people in parallel, in which each person works in a scale.

The multiscale approach led to new challenges:

**Update propagation among scales:** Since in our strategy the integration is sliced in several scales, changes on the underlying sources should be propagated to the dataspace, implying in changes on subsequent scales.

**Versioning:** Within a dataspace, links may be produced or destroyed over time. Such changes could be versioned, providing to the dataspace the capability of restoring a previous state.



**Tighter integration with Network related tools:** The results produced by Linked-Scales can feed graph-oriented external tools like: network topology analysis tools and knowledge network tools.

**Full-fledged implementation with a graphical interface:** Several aspects of the proposal were implemented, including the scale-specialized querying environments and the transformation services. Conciliating such implemented solutions in a single tool that hides technical aspects behind a graphical interface may allow users to produce better integrated views of the data.

# Bibliography

- [1] Renzo Angles. A comparison of current graph database models. In *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*, pages 171–177. IEEE, 2012.
- [2] Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Computing Surveys*, 40(1):1:1–1:39, February 2008.
- [3] Pablo Barceló, Leonid Libkin, and Juan L. Reutter. Querying Regular Graph Patterns. *Journal of the ACM*, 61(1):1–54, jan 2014.
- [4] Pablo Barceló Baeza. Querying graph databases. In *Proceedings of the 32nd symposium on Principles of database systems - PODS '13*, volume 1777, page 175, New York, New York, USA, 2013. ACM Press.
- [5] Khalid Belhajjame, Norman W. Paton, Suzanne M. Embury, Alvaro A. A. Fernandes, and Cornelia Hedeler. Feedback-based annotation, selection and refinement of schema mappings for dataspace. In *Proceedings of the 13th International Conference on Extending Database Technology, EDBT '10*, pages 573–584, New York, NY, USA, 2010. ACM.
- [6] Khalid Belhajjame, Norman W. Paton, Suzanne M. Embury, Alvaro A.A. Fernandes, and Cornelia Hedeler. Incrementally improving dataspace based on user feedback. *Information Systems*, 38(5):656 – 687, 2013.
- [7] Gordon Bell, Tony Hey, and Alex Szalay. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.
- [8] Ivelize R Bernardo, Matheus Silva Mota, and André Santanchè. Extracting and semantically integrating implicit schemas from multiple spreadsheets of biology based on the recognition of their nature. *Journal of Info. and Data Manag.*, 4(2):104, 2013.
- [9] Farida A Bhuiya, Stephen R Pitts, and Linda F McCaig. Emergency department visits for chest pain and abdominal pain: United states, 1999-2008. *NCHS data brief*, (43):1–8, 2010.
- [10] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

- [11] Tara Borlawsky, Carol Friedman, and Yves a Lussier. Generating executable knowledge for evidence-based medicine using natural language and semantic processing. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 56–60, 2006.
- [12] David Guy Brizan and Abdullah Uz Tansel. A. survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, 6(3):5, 2006.
- [13] Bruce Buchanan and Edward Shortliffe. Rule-based expert systems: the mycin experiments of the stanford heuristic programming project. 1984.
- [14] Yuhan Cai, Xin Luna Dong, Alon Halevy, Jing Michelle Liu, and Jayant Madhavan. Personal information management with semex. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 921–923, New York, NY, USA, 2005. ACM.
- [15] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y Vardi. Rewriting of regular expressions and regular path queries. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 194–204. ACM, 1999.
- [16] Sanjeev D Chunilal, John W Eikelboom, John Attia, Massimo Miniati, Akbar A Panju, David L Simel, and Jeffrey S Ginsberg. Does this patient have pulmonary embolism? *Jama*, 290(21):2849–2858, 2003.
- [17] Isabel F. Cruz, Alberto O. Mendelzon, and Peter T. Wood. A graphical query language supporting recursion. *ACM SIGMOD Record*, 16(3):323–330, December 1987.
- [18] Tiago de Araujo Guerra Grangeia, Bruno de Jorge, Daniel Franci, Thiago Martins Santos, Maria Silvia Vellutini Setubal, Marcelo Schweller, and Marco Antonio de Carvalho-Filho. Cognitive Load and Self-Determination Theories Applied to E-Learning: Impact on Students' Participation and Academic Performance. *PLOS ONE*, 11(3):e0152462, mar 2016.
- [19] Jens Dittrich, Marcos Antonio Vaz Salles, and Lukas Blunski. imemex: From search to information integration and back. *IEEE Data Eng. Bull.*, 32(2):28–35, 2009.
- [20] Jens-Peter Dittrich and Marcos Antonio Vaz Salles. idm: A unified and versatile data model for personal dataspace management. In *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, pages 367–378. VLDB Endowment, 2006.
- [21] Benjamin Djulbegovic and Gordon H Guyatt. Progress in evidence-based medicine: a quarter century on. *The Lancet*, 6736(16):1–9, feb 2017.
- [22] Xin Dong and Alon Halevy. Indexing dataspaces. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 43–54, New York, NY, USA, 2007. ACM.

- [23] I. Elsayed, P. Brezany, and A.M. Tjoa. Towards realization of dataspaces. In *17th International Conference on Database and Expert Systems Applications (DEXA 06)*, pages 266–272. IEEE, 2006.
- [24] Ibrahim Elsayed and Peter Brezany. Towards large-scale scientific dataspaces for e-science applications. In *Database Systems for Advanced Applications*, pages 69–80. Springer, 2010.
- [25] Gary N Fox and Nashat S Moawad. Uptodate: a comprehensive clinical database. *Journal of family practice*, 52(9):706–710, 2003.
- [26] Michael Franklin, Alon Halevy, and David Maier. From databases to dataspaces: a new abstraction for information management. *ACM Sigmod Record*, 34(4), 2005.
- [27] Lise Getoor and Ashwin Machanavajjhala. Entity resolution for big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1527–1527. ACM, 2013.
- [28] L.M. Haas, E. T. Lin, and M. A. Roth. Data integration through database federation. *IBM Systems Journal*, 41(4):578–596, 2002.
- [29] Alon Halevy, Michael Franklin, and David Maier. Principles of dataspace systems. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS*, PODS '06, pages 1–9, New York, NY, USA, 2006. ACM.
- [30] Alon Halevy, Anand Rajaraman, and Joann Ordille. Data integration: The teenage years. In *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, pages 9–16. VLDB Endowment, 2006.
- [31] Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 2011.
- [32] Cornelia Hedeler, Khalid Belhajjame, AlvaroA.A. Fernandes, SuzanneM. Embury, and NormanW. Paton. Dimensions of dataspaces. In AlanP. Sexton, editor, *Dataspace: The Final Frontier*, volume 5588 of *Lecture Notes in Computer Science*, pages 55–66. Springer Berlin Heidelberg, 2009.
- [33] Cornelia Hedeler, Khalid Belhajjame, Norman W. Paton, Alvaro A.A. Fernandes, Suzanne M. Embury, Lu Mao, and Chenjuan Guo. Pay-as-you-go mapping selection in dataspaces. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 1279–1282, New York, NY, USA, 2011. ACM.
- [34] Cornelia Hedeler, Khalid Belhajjame, NormanW. Paton, Alessandro Campi, AlvaroA.A. Fernandes, and SuzanneM. Embury. Chapter 7: Dataspaces. In Stefano Ceri and Marco Brambilla, editors, *Search Computing*, volume 5950 of *Lecture Notes in Computer Science*, pages 114–134. Springer Berlin Heidelberg, 2010.

- [35] Cornelia Hedeler, Alvaro A. A. Fernandes, Khalid Belhajjame, Lu Mao, Chenjuan Guo, Norman W. Paton, and Suzanne M. Embury. A functional model for dataspace management systems. In Barbara Catania and Lakhmi C. Jain, editors, *Advanced Query Processing*, volume 36 of *Intelligent Systems Reference Library*, pages 305–341. Springer Berlin Heidelberg, 2013.
- [36] S Blair Hedges. The origin and evolution of model organisms. *Nature Reviews Genetics*, 3(11):838–849, 2002.
- [37] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington, 2009.
- [38] Tammy C Hoffmann, Victor M Montori, and Chris Del Mar. The connection between evidence-based medicine and shared decision making. *Jama*, 312(13):1295–1296, 2014.
- [39] Florian Holzschuher and René Peinl. Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 195–204. ACM, 2013.
- [40] Shawn R. Jeffery, Michael J. Franklin, and Alon Y. Halevy. Pay-as-you-go user feedback for dataspace systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 847–860, New York, NY, USA, 2008. ACM.
- [41] Salim Jouili and Valentin Vansteenberghe. An empirical comparison of graph databases. In *Social Computing (SocialCom), 2013 International Conference on*, pages 708–715. IEEE, 2013.
- [42] Alan Jović, Dragan Gamberger, and Goran Krstajić. Heart failure ontology. *Bio-algorithms and med-systems*, 7(2):101–110, 2011.
- [43] Jeffrey A Kline and Christopher Kabrhel. Emergency evaluation for pulmonary embolism, part 1: clinical factors that increase risk. *The Journal of emergency medicine*, 48(6):771–780, 2015.
- [44] Michael Klompas. Does This Patient Have an Acute Thoracic Aortic Dissection? *JAMA*, 287(17):2262, may 2002.
- [45] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets semantic web—how the bbc uses dbpedia and linked data to make connections. In *The semantic web: research and applications*, pages 723–737. Springer, 2009.
- [46] Phokion G. Kolaitis. Schema mappings, data exchange, and metadata management. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '05, pages 61–75, New York, NY, USA, 2005. ACM.

- [47] Evangelos Kontopantelis, David A Springate, and David Reeves. A re-analysis of the cochrane library data: the dangers of unobserved heterogeneity in meta-analyses. *PloS one*, 8(7):e69930, 2013.
- [48] Casimir A Kulikowski and Sholom M Weiss. Representation of expert knowledge for consultation: the CASNET and EXPERT projects. *Artificial Intelligence in medicine*, 51, 1982.
- [49] Timothy Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. Prov-o: The prov ontology. *W3C Recommendation*, 30, 2013.
- [50] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 233–246, New York, NY, USA, 2002. ACM.
- [51] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [52] Graeme T. Lloyd, Steve C. Wang, and Stephen L. Brusatte. Identifying heterogeneity in rates of morphological evolution: Discrete character change in the evolution of lungfish (sarcopterygii; dipnoi). *Evolution*, 66(2):330–348, 2012.
- [53] Jayant Madhavan, Shirley Cohen, Xin Luna Dong, Alon Y. Halevy, Shawn R. Jeffery, David Ko, and Cong Yu. Web-scale data integration: You can afford to pay as you go. In *CIDR*, pages 342–350, 2007.
- [54] Steven McGee. *Evidence-Based Physical Diagnosis*. Elsevier Inc., 2012.
- [55] Eduardo Miranda and André Santanchè. Unifying phenotypes to support semantic descriptions. *Brazilian Conference on Ontological Research – ONTOBRAS*, pages 1–12, October 2013.
- [56] Matheus Silva Mota, Júlio Cesar dos Reis, Sandra Goutte, and André Santanchè. Multiscaling a graph-based dataspace. *Journal of Information and Data Management - JIDM*, 7(3):233–248, 2016.
- [57] Matheus Silva Mota, Julio Cesar dos Reis, Sandra Goutte, and André Santanchè. Multiscale dataspace for organism-centric analysis. In *Proceedings of the Brazilian Symposium on Databases (SBBD)*, 2015.
- [58] Matheus Silva Mota, João Sávio C Longo, Daniel C Cugler, and Claudia Bauzer Medeiros. Using linked data to extract geo-knowledge. In *GeoInfo*, pages 111–116, 2011.
- [59] Matheus Silva Mota and Claudia Bauzer Medeiros. Introducing shadows: Flexible document representation and annotation on the web. In *Proc. of Data Engineering Workshops (ICDEW), IEEE 29th ICDE*, pages 13–18. IEEE, apr 2013.

- [60] Matheus Silva Mota, Fagner Leal Pantoja, Júlio Cesar dos Reis, and André Santanchè. Progressive data integration and semantic enrichment based on linked scales and trails. In *Proceedings of the 9th International Conference Semantic Web Applications and Tools for Life Sciences (swat4ls)*, 2016.
- [61] Matheus Silva Mota and André Santanchè. Conceiving a multiscale dataspace for data analysis. In Brazilian Conference on Ontologies (Ontobras), editor, *Proceedings of the Brazilian Seminar on Ontologies (ONTOBRAS 2015)*, volume 1442, page 12. CEUR, 2015.
- [62] Seán I O’Donoghue, Anne-Claude Gavin, Nils Gehlenborg, David S Goodsell, Jean-Karim Hériché, Cydney B Nielsen, Chris North, Arthur J Olson, James B Procter, David W Shattuck, et al. Visualizing biological data—now and in the future. *Nature methods*, 7:S2–S4, 2010.
- [63] Shraddha Pandit and Suchita Gupta. A comparative study on distance measuring approaches for clustering. *International Journal of Research in Computer Science*, 2(1):29, 2011.
- [64] Akbar A Panju, Brenda R Hemmelgarn, Gordon H Guyatt, and David L Simel. Is this patient having a myocardial infarction? *Jama*, 280(14):1256–1263, 1998.
- [65] Fagner Leal Pantoja, Patrícia Cavoto, Julio Reis, and André Santanchè. Generating Knowledge Networks from Phenotypic Descriptions. *Proc. 12th IEEE e-Science*, pages 1–10, 2016.
- [66] Jason D. Pardo, Adam K. Huttenlocker, and Bryan J. Small. An exceptionally preserved transitional lungfish from the lower permian of nebraska, usa, and the origin of modern lungfishes. *PLoS ONE*, 9(9):1–13, 09 2014.
- [67] Norman W. Paton, Klitos Christodoulou, Alvaro A. A. Fernandes, Bijan Parsia, and Cornelia Hedeler. Pay-as-you-go data integration for linked data: opportunities, challenges and architectures. In *Proceedings of the 4th International Workshop on Semantic Web Information Management, SWIM ’12*, pages 3:1–3:8, New York, NY, USA, 2012. ACM.
- [68] Peter S. Rahko. Rapid evaluation of chest pain in the emergency department. *JAMA Internal Medicine*, 174(1):59–60, jan 2014.
- [69] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
- [70] W. Rosenberg and A. Donald. Evidence based medicine: an approach to clinical problem-solving. *Bmj*, 310(6987):1122–1126, apr 1995.
- [71] David L. Sackett. Evidence-based medicine. *Seminars in Perinatology*, 21(1):3–5, feb 1997.

- [72] R Sanderson, P Ciccarese, and B Young. Web annotation data model. w3c recommendation, 23 february 2017, 2017.
- [73] Allen F Shaughnessy, John R Torro, Kara A Frame, and Munish Bakshi. Evidence-based medicine and life-long learning competency requirements in new residency teaching standards. *Evidence-based medicine*, 21(2):46–49, 2016.
- [74] M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann, and G. F. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30(4):241–255, 1991.
- [75] Mrityunjay Singh and S. K. Jain. A survey on dataspace. In DavidC. Wyld, Michal Wozniak, Nabendu Chaki, Natarajan Meghanathan, and Dhinaharan Nagamalai, editors, *Advances in Network Security and Applications*, volume 196 of *Communications in Computer and Information Science*, pages 608–621. Springer Berlin Heidelberg, 2011.
- [76] Peter Szolovits and Lucila Ohno-Machado. Updating the QMR in 2005: New Approaches. 2005.
- [77] Jan van Dijk, Sunil Choenni, Erik Leertouwer, Marco Spruit, and Sjaak Brinkkemper. A data space system for the criminal justice chain. In Robert Meersman, Herve Panetto, Tharam Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter Leenheer, and Deijing Dou, editors, *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, volume 8185 of *Lecture Notes in Computer Science*, pages 755–763. Springer Berlin Heidelberg, 2013.
- [78] Marcos Antonio Vaz Salles, Jens-Peter Dittrich, Shant Kirakos Karakashian, Olivier René Girard, and Lukas Blunschi. itrails: Pay-as-you-go information integration in dataspaces. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, pages 663–674. VLDB Endowment, 2007.
- [79] Robert A Wagner and Michael J Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.
- [80] Liqin Wang, Bruce E Bray, Jianlin Shi, Guilherme Del Fiol, and Peter J Haug. A method for the development of disease-specific reference standards vocabularies from textual biomedical literature resources. *Artificial intelligence in medicine*, 68:47–57, 2016.
- [81] Nicole L Washington, Melissa A Haendel, Christopher J Mungall, Michael Ashburner, Monte Westerfield, and Suzanna E Lewis. Linking human diseases to animal models using ontologybased phenotype annotation. *PLoS biology*, 7(11):e1000247, 2009.
- [82] William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. 1990.



- [83] Peter T Wood. Query languages for graph databases. *ACM SIGMOD Record*, 41(1):50, apr 2012.
- [84] Andrew Worster and Teresa Chan. Does this patient have a hemorrhagic stroke? *Annals of Emergency Medicine*, 57(5):535–536, may 2011.
- [85] Dan Yang, Derong Shen, Tiezheng Nie, Ge Yu, and Yue Kou. Layered graph data model for data management of dataspace support platform. In Haixun Wang, Shijun Li, Satoshi Oyama, Xiaohua Hu, and Tieyun Qian, editors, *Web-Age Information Management*, volume 6897 of *Lecture Notes in Computer Science*, pages 353–365. Springer Berlin Heidelberg, 2011.
- [86] Nazar Zaki, Chandana Tennakoon, and Hany Al Ashwal. Knowledge graph construction and search for biological databases. In *Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on*, pages 1–6. IEEE, 2017.
- [87] Ming Zhong, Mengchi Liu, and Yanxiang He. 3sepia: A semi-structured search engine for personal information in dataspace system. *Information Sciences*, 218(0):31 – 50, 2013.

# Appendix A

## Conceiving a Multiscale Dataspace for Data Analysis

### A.1 Introduction and Motivation

From science to business, several domains are facing a huge increase in the amount of available data and the growth of the data heterogeneity (in various levels). In parallel, opportunities may emerge from the exploitation of the increasing volume of connections among multidisciplinary data [37].

Domains like biology are increasingly becoming data-driven. Although they adopt different systems to produce, store and search their data, biologists increasingly need a unified view of these data to understand and discover relationships between low-level (e.g., cellular, genomic or molecular level) and high-level (e.g., species characterization, macro-biomass etc.) biological information among several heterogeneous and distributed sources. Therefore, integration becomes a key factor in such data-intensive and in multidisciplinary domains; the production and exploitation of connections among independent data-sources become essential [24]. Besides integration, challenges like provenance, visualization and versioning are experienced by domains that handle large, heterogeneous and cross-connected datasets [31].

In order to integrate available sources, classical data integration approaches, found in the literature, usually require an up-front effort related to schema recognition/mapping in an all-or-nothing fashion [29]. On demand integration of distinct and heterogeneous sources requires ad hoc solutions and repeated effort from specialists [26].

*Franklin et al.* propose the notion of *dataspaces* to address the problems mentioned above [26]. The dataspace vision aims to provide the benefits of the classical data integration approach, but via a progressive “pay-as-you-go” integration [29]. They argue that linking lots of “fine-grained” information particles, bearing “little semantics”, already bring benefits to applications, and more links can be produced on demand, as *lightweight* steps of integration.

Related work proposals address distinct aspects of dataspace. Regarding the architectural aspect, each work explores a different issue of a dataspace system. Among all efforts, no dominant proposal of a complete architecture has emerged until now. We

observed that, in a progressive integration process, steps are not all alike. They can be distinguished by interdependent roles, which we organize here as abstraction layers. They are materialized in our LinkedScales, a graph-based dataspace architecture. Inspired by a common backbone found in related work, LinkedScales aims to provide an architecture for dataspace systems that supports progressive integration and the management of heterogeneous sources.

LinkedScales takes advantage of the flexibility of graph structures and proposes the notion of scales of integration. Scales are represented as graphs, managed in graph databases. Operations become transformations of such graphs. LinkedScales also systematically defines a set of scales as layers, where each scale focuses in a different level of integration and its respective abstraction. In a progressive integration, each scale congregates homologous lightweight steps. They are interconnected, supporting provenance traceability. Furthermore, LinkedScales supports a complete dataspace lifecycle, including automatic initialization, maintenance and refinement of the links.

This paper discusses the conceiving of the LinkedScales architecture and is organized as follows. Section A.2 discusses some concepts and related work. Section A.3 introduces the LinkedScales proposal, also discussing previous work and how such experiences led to the proposed architecture. Section A.4 presents previous work on data integration and discusses how such experiences are reflected in current proposal. Finally, Section A.5 presents some conclusions and future steps.

## A.2 Related Work

### A.2.1 The Classical Data Integration

Motivated by such increasingly need of treating multiple and heterogeneous data sources, data integration has been the focus of attention in the database community in the past two decades [35]. One predominant strategy is based on providing a virtual unified view under a global schema (GS) [46]. Within GS systems, the data stay in their original data sources – maintaining their original schemas – and are dynamically fetched and mapped to a global schema under clients’ request [50, 35]. In a nutshell, applications send queries to a mediator, which maps them into several sub-queries dispatched to wrappers, according to metadata regarding capabilities of the participating DBMSs. Wrappers map queries to the underlying DBMSs and the results back to the mediator, guided by the global schema. Queries are optimized and evaluated according to each DBMS within the set, providing the illusion of a single database to applications [50].

A main problem found in this “classical” data integration strategy regards the big upfront effort required to produce a global schema definition [30]. Since in some domains different DBMSs may emerge and schemas are constantly changing, such costly initial step can become impracticable [35]. Moreover, several approaches focus on a particular data model (e.g., relational), while new models also become popular [23]. As we will present in next section, an alternative to this classical all-or-nothing costly upfront data integration strategy is a strategy based on progressive small integration steps.

### A.2.2 The “Pay-as-you-go” Dataspace Vision

Since upfront mapping between schemas are labor intensive and scheme-static domains are rare, pay-as-you-go integration strategies have gained momentum. Classical data integration (presented in Section A.2.1) approaches work successfully when integrating modest numbers of stable databases in controlled environments, but lack an efficient solution for scenarios in which schemas often change and new data models must be considered [35]. In a data integration spectrum, the classical data integration is at the high-cost/high-quality end, while an incremental integration based on progressive small steps starts in the opposite side. However, this incremental integration can be continuously refined in order to improve the connections among sources.

In 2005, *Franklin et al.* published a paper proposing the notion of *dataspaces*. The dataspace vision aims at providing the benefits of the classical data integration approach, but in a progressive fashion [29, 75, 34]. The main argument behind the dataspace proposal is that, in the current scenario, instead of a long wait for a global integration schema to have access to the data, users would rather to have early access to the data, among small cycles of integration – i.e., if the user needs the data now, some integration is better than nothing. This second generation approach of data integration can be divided in a bootstrapping stage and subsequent improvements. Progressive integration refinements can be based, for instance, on structural analysis [22], on user feedback [6] or on manual / automatic mappings among sources – if benefits worth such effort.

Dataspaces comprise several challenges related to the design of Dataspace Support Platforms (DSSPs). The main goal of a DSSP is to provide basic support for operations among all data sources within a dataspace, allowing developers to focus on specific challenges of their applications, rather than handling low-level tasks related to data integration [75]. Many DSSPs have been proposed recently addressing a variety of scenarios, e.g., SEMEX [14] and iMeMex [19] on the PIM context; PayGo [53] focusing on Web-related sources; and a justice-related DSSP[77]. As far as we know, up to date, the proposed DSSPs provide specialized solutions, targeting only specific scenarios [75, 32].

## A.3 LinkedScales: A Multiscale Dataspace Architecture

The goal of LinkedScales is to systematize the dataspace-based integration process in an architecture. It slices integration levels in progressive layers, whose abstraction is inspired by the notion of scales. As an initial effort, LinkedScales strategy focuses on a specific goal on the dataspace scope: to provide a homogeneous view of data, hiding details about heterogeneous and specific formats and schemas. To achieve this goal, the current proposal does not address issues related to access policies, broadcast updates or distributed access management.

LinkedScales is an architecture for systematic and incremental data integration, based on graph transformations, materialized in different scales of abstraction. It aims to support algorithms and common tools for integrating data within the dataspaces. Integration-

scales are linked, and data in lower scales are connected to their corresponding representations in higher scales. As discussed in next section, each integration-scale is based on experiences acquired in three previous experiences related to data integration.

Figure A.1 shows an overview of the LinkedScales DSSP architecture, presenting, from bottom to top the following scales of abstraction. (i) *Physical Scale*, (ii) *Logical Scale*; (iii) *Description Scale*; and (iv) *Conceptual Scale*.

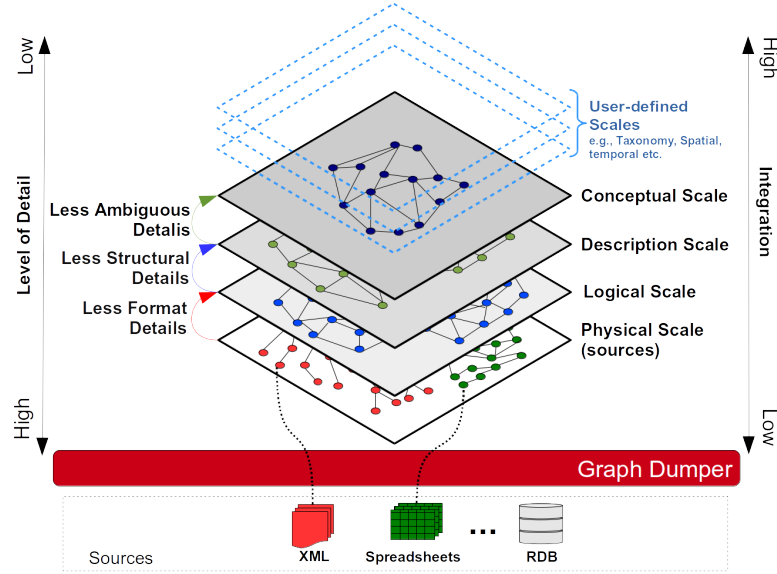


Figure A.1: Overview of the LinkedScales architecture.

The lowest part of Figure A.1 – the *Graph Dumper* and the *Sources* – represents the different data sources handled by our DSSP in their original format. Even though we are conceiving an architecture that can be extended to any desired format, we are currently focusing on spreadsheets, XML files and textual documents as underlying sources. Data at this level are treated as black-boxes. Therefore, data items inside the sources are still not addressable by links.

The lower scale – the *Physical Scale* – aims at mapping the sources available in the dataspace to a graph inside a graph database. This type of database stores graphs in their native model and they are optimized to store and handle them. The operations and query languages are tailored for graphs. There are several competing approaches to represent graphs inside the database [1, 2].

The *Physical Scale* is the lowest-level raw content+format representation of data sources with addressable/linkable component items. It will reflect in a graph, as far as possible, the original structure and content of the original underlying data sources. The role of this scale – in an incremental integration process – concerns making explicit and linkable data within sources. In a dataspace fashion, such effort to make raw content explicit can be improved on demand.

The *Logical Scale* aims at offering a common view to data inside similar or equivalent structural models. Examples of structural models are: table and hierarchical document. In the previous scale, there will be differences in the representation of a table within a PDF, a table from a spreadsheet and a table within a HTML file, since they preserve

specificities of their formats. In this (Logical) scale, on the other hand, the three tables should be represented in the same fashion, since they refer to the same structural model. This will lead to a homogeneous approach to process tables, independently of how tables were represented in their original specialized formats. To design the structural models of the Logical Scale we will investigate initiatives such as the OMG's<sup>1</sup> Information Management Metamodel<sup>2</sup> (IMM). IMM addresses the heterogeneity among the models behind Information Management systems, proposing a general interconnected metamodel, aligning several existing metamodels. Figure A.2 presents an overview of the current state of the IMM and supported metamodels. For instance, it shows that XML and Relational metamodels can be aligned into a common metamodel.

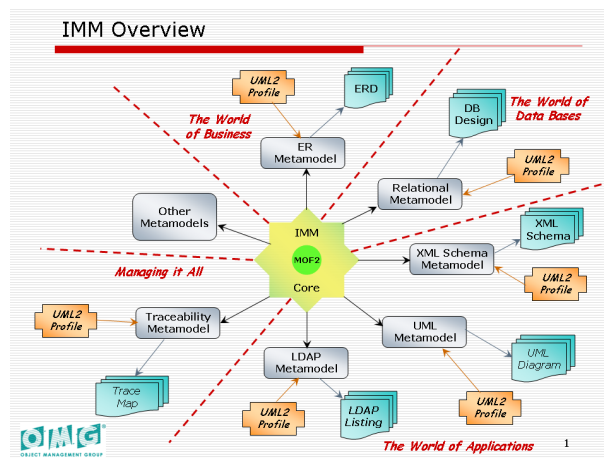


Figure A.2: Overview of the current state of the IMM. Source: [www.omgwiki.org/imm](http://www.omgwiki.org/imm)

In the *Description Scale*, the focus is in the content (e.g., labels of tags within a XML or values in spreadsheet cells) and their relationships. Structural information pertaining to specific models – e.g., aggregation nodes of XML – are discarded if they do not affect the semantic interpretation of the data, otherwise, they will be transformed in a relation between nodes following common patterns – for example, cells in the same row of a table are usually values for attributes of a given entity. Here, the structures from previous scales will be reflected as RDF triples.

The highest scale of Figure A.1 is the *Conceptual Scale*. It unifies in a common semantic framework the data of the lower scale. Algorithms to map content to this scale exploit relationships between nodes of the Description Scale to discover and to make explicit as ontologies the latent semantics in the existing content. As we discuss in next section, it is possible in several scenarios to infer semantic entities – e.g., instances of classes in ontologies – and their properties from the content. We are also considering the existence of predefined ontologies, mapped straight to this scale, which will support the mapping process and will be connected to the inferred entities. Here, algorithms concerning entity linking should be investigated.

<sup>1</sup>Object Management Group – <http://www.omg.org>

<sup>2</sup><http://www.omgwiki.org/imm>

## A.4 Previous Work

This proposal was conceived after experiences acquired during three previous research projects. Although with different strategies, they addressed complementary issues concerning data integration. In each project, experiments were conducted in a progressive integration fashion, starting from independent artifacts – represented by proprietary formats, in many cases – going towards the production of connections in lightweight or heavyweight integration approaches. As we will show here, our heavyweight integration here took a different perspective from an upfront one-step integration. It is the end of a chain of integration steps, in which the semantics inferred from the content in the first integration steps influences the following integration steps.

We further detail and discuss the role of each work in the LinkedScales architecture. While [59] explores a homogeneous representation model for textual documents independently of their formats, [8] and [55] focus, respectively, on extracting and recognizing relevant information stored in spreadsheets and XML artifacts, to exploit their latent semantics in integration tasks.

### A.4.1 Homogeneous Model – Universal Lens for Textual Document Formats

One of the key limits to index, handle, integrate and summarize sets of documents is the heterogeneity of their formats. In order to address this problem, we envisaged a “document space” in which several document sources represented in heterogeneous formats are mapped to a homogeneous model we call *Shadow* [59].

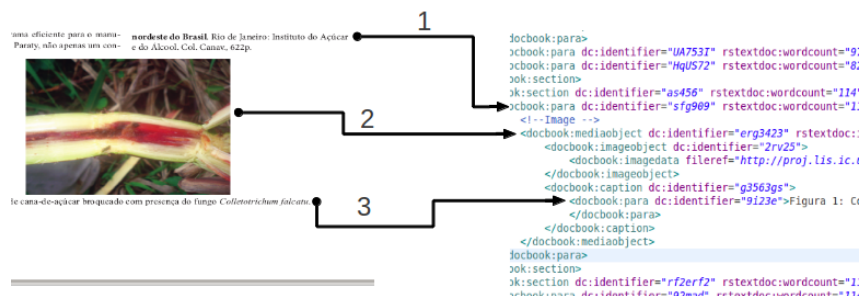


Figure A.3: Main idea behind the work [59]: A PDF document and its corresponding shadow.

Figure A.3 illustrates a typical Shadow (serialized in XML). The content and structure of a document in a specific format (e.g., PDF, ODT, DOC) is extracted and mapped to an open structure – previously defined. The model behind this new structure, which is homogeneous across documents in the space, is a common hierarchical denominator found in most textual documents – e.g., sections, paragraphs, images. In the new document space a shadow represents *format+structure* of a document, decoupled from its specialized format.

Shadows documents are abstractions of documents in specific formats, i.e., they do not represent integrally the information of the source, focusing in the common information

that can be extracted according to the context. This abstract homogeneous model allowed us to develop interesting applications in: document content integration and semantic enrichment [58]; and searching in a document collection considering structural elements, such as labels of images or references [59].

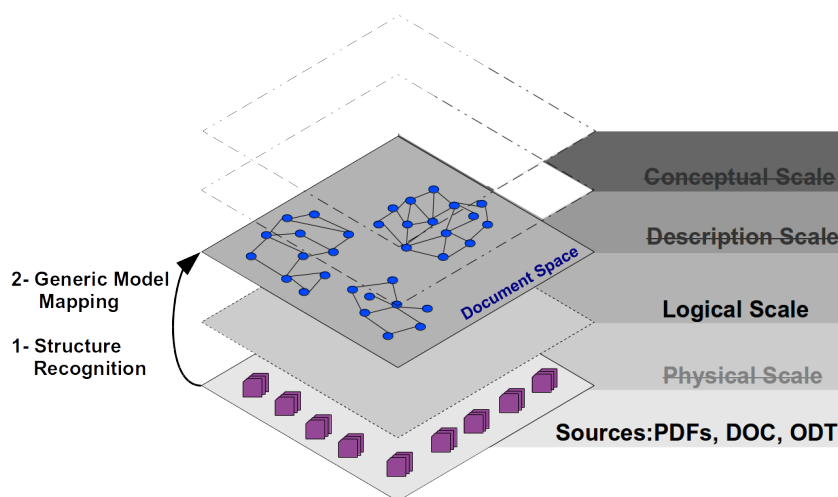


Figure A.4: Shadows approach presented in a LinkedScales perspective.

Figure A.4 illustrates how this homogeneous view for a document space fits in the LinkedScales architecture. This document space is equivalent to the Logical Scale, restricted to the document context. Different from the LinkedScales approach, Shadows map the documents in their original format straight to the generic model, without an intermediary Physical Scale.

After the Shadows experience we observed three important arguments to represent such intermediary scale: (i) since this scale is not aimed at mapping the resources to a common model, it focus in the specific concern of making explicit and addressable the content; (ii) it preserves the best-effort graph representation of the source, with provenance benefits; (iii) the big effort in the original one-batch-way conversion is factored in smaller steps with intermediary benefits.

In the LinkedScales' *Logical Scale*, the Shadows' document-driven common model will be expanded towards a generic perspective involving a family of models.

#### A.4.2 Connecting descriptive XML data – a Linked Biology perspective

[55] studied a particular problem in the biology domain, related to phenotypic descriptions and their relations with phylogenetic trees. Phenotypic descriptions are a fundamental starting point for several biology tasks, like identification of living beings or phylogenetic tree construction. Tools for this kind of description usually store data in independent files following open standards (e.g., XML). The descriptions are still based on textual sentences in natural language, limiting the support of machines in integration, correlation and comparison operations.

Even though modern phenotype description proposals are based on ontologies, there



still are open problems of how to take advantage of the existing patrimony of descriptions. In such scenario, [55] proposes a progressive integration approach based on successive graph transformations, which exploits the existing latent semantics in the descriptions to guide this integration and semantic enrichment.

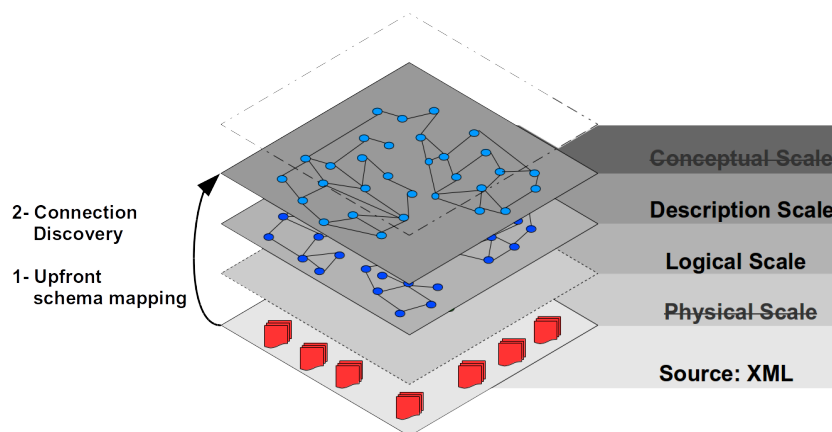


Figure A.5: Linked Biology project presented in a LinkedScales perspective.

Since the focus is in the content, this approach departs from a graph-based schema which is a minimal common denominator among the main phenotypic description standards. Operations which analyses the content – discovering hidden relations – drive the integration process. Figure A.5 draws the intersection between our architecture and the integration approach proposed by [55]. Data of the original artifacts are mapped straight to the Description Scale, in which structures have a secondary role and the focus is in the content.

In spite of the benefits of the focus in the content, simplifying the structures, this approach loses information which will be relevant for provenance. Moreover, in an interactive integration process, the user can perceive the importance of some information not previously considered in the Description Scale. In this case, since the mapping comes straight from the original sources, it becomes a hard task to update the extraction/mapping algorithms to afford each new requirement. The Physical and Logical Scales simplify this interactive process, since new requirements means updating graph transformations from lower to upper scales.

### A.4.3 Progressively Integrating Biology Spreadsheet Data

Even though spreadsheets play important role as “popular databases”, they were designed as self contained units. This characteristic becomes an obstacle when users need to integrate data from several spreadsheets, since the content is strongly coupled to file formats, and schemas are implicit driven to human consumption. In [8], we decoupled the content from the structure to discover and make explicit the implicit schema embedded in the spreadsheets.

Figure A.6 illustrates the [8] approach in a LinkedScales perspective. The work is divided in four steps, going from the original spreadsheets formats straight to the *Conceptual Scale*. The first step is to recognize the spreadsheet nature. The work assumes

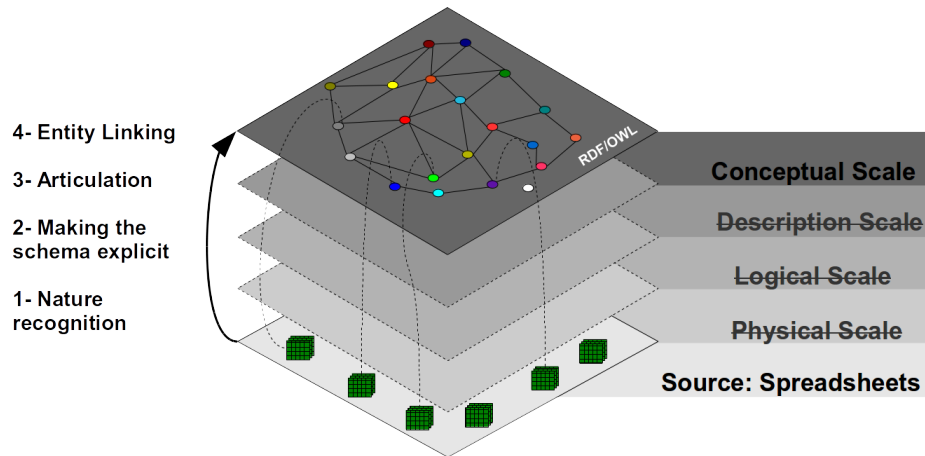


Figure A.6: Spreadsheet integration presented in a LinkedScales perspective.

that users follow and share domain-specific practices when they are constructing spreadsheets, which result in patterns to build them. Such patterns are exploited in order to capture the nature of the spreadsheet and to infer a conceptual model behind the pattern, which will reflect in an ontology class in the *Conceptual Scale*.

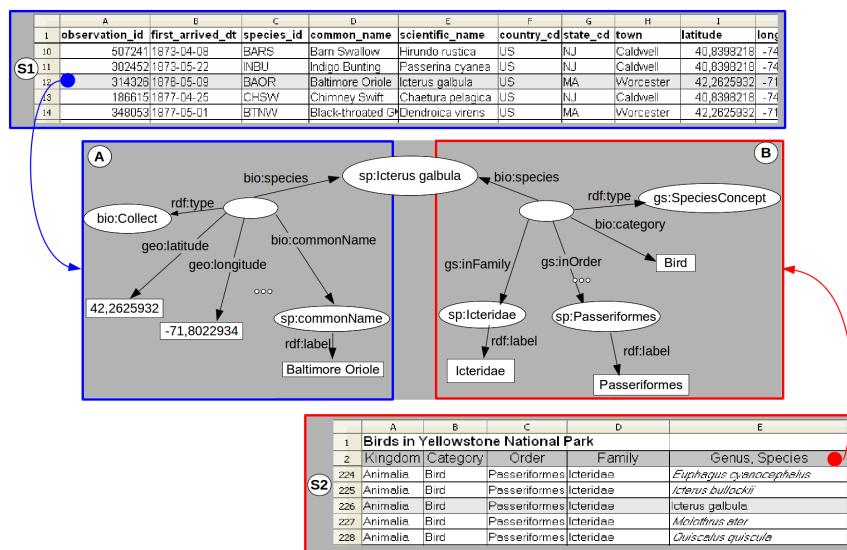


Figure A.7: Spreadsheet data articulation via entity recognition.

This work stresses the importance of recognizing data as semantic entities to guide further operations of integration and articulation. Via this strategy, authors are able to transform several spreadsheets into a unified and integrated data repository. Figure A.7 shows an example summarizing how they are articulated, starting from the recognition of semantic entities behind implicit schemas. Two different spreadsheets (*S1* and *S2*) related to the biology domain have their schema recognized and mapped to specific ontology classes – shown in Figure A.7 as (A) and (B).

Semantic entities can be properly interpreted, articulated and integrated with other sources – such as DBpedia, GeoSpecies and other open datasets. In an experiment involving more than 11,000 spreadsheets, we showed that it is possible to automatically recognize and merge entities extracted from several spreadsheets.

Figure A.8 shows a screencopy of our query and visualization prototype for data<sup>3</sup> extracted from spreadsheets (available in <http://purl.org/biospread/?task=pages/txnavigator>).

This work subsidized our proposal of a Conceptual Scale as the topmost layer of our LinkedScales architecture. Several intermediary steps of transformation from the original datasources towards entities are hidden inside the extraction/mapping program. As in the previous cases, the process can be improved by materializing these intermediate steps in scales of our architecture.

## A Taxonomy Navigator

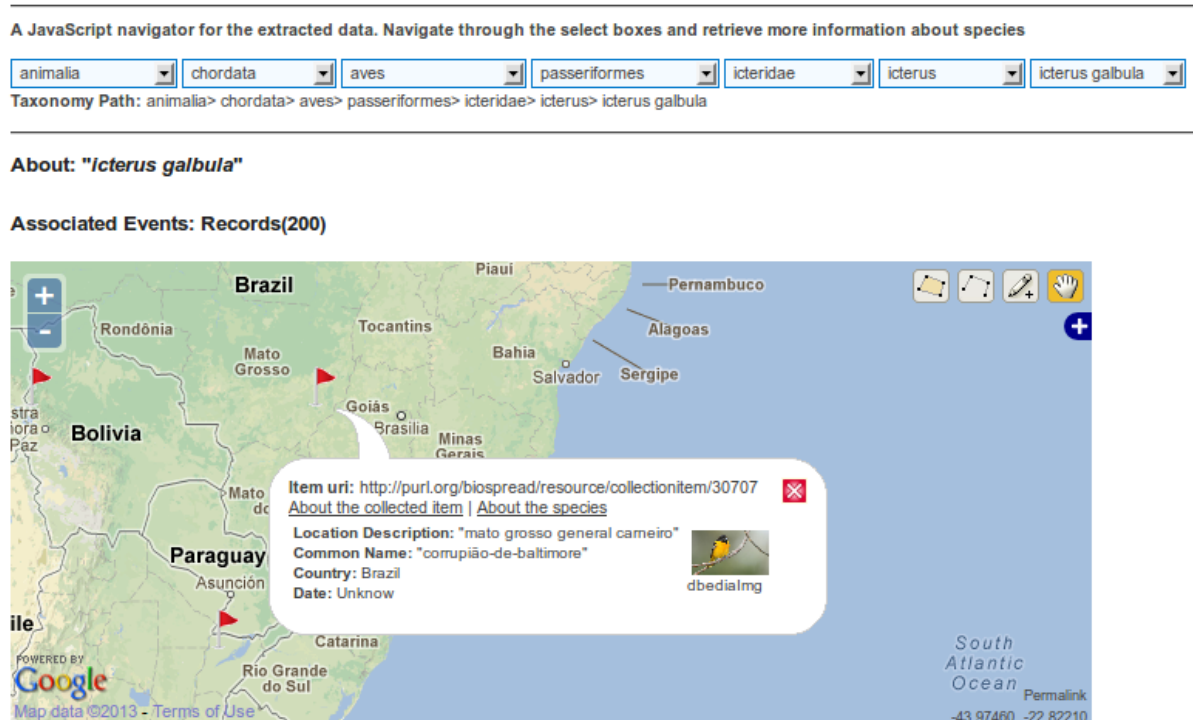


Figure A.8: Screencopy of our prototype integrating data of several spreadsheets.

## A.5 Concluding Remarks

This work presented a proposal for a dataspace system architecture based on graphs. It systematizes in layers (scales) progressive integration steps, based in graph transformations. The model is founded in previous work, which explored different aspects of the proposal. LinkedScales is aligned with the modern perspective of treating several heterogeneous datasources as parts of the same dataspace, addressing integration issues in progressive steps, triggered on demand. Although our focus is in the architectural aspects, we are designing a generic architecture able to be extended to several contexts.

<sup>3</sup>All data is available at our SPARQL endpoint: <http://sparql.lis.ic.unicamp.br>